E.T.S. INGENIEROS DE CAMINOS, CANALES Y PUERTOS

DEPARTAMENTO DE MÉTODOS MATEMÁTICOS Y DE REPRESENTACIÓN

Doctoral Thesis:

An immersed methodology for fluid-structure interaction using NURBS and T-splines: Theory, algorithms, validation, and application to blood flow at small scales

Submitted by Hugo Casquero Penelas

Supervised by:

Héctor Gómez Díaz

A Coruña - June 2016

E.T.S. INGENIEROS DE CAMINOS, CANALES Y PUERTOS

DEPARTAMENTO DE MÉTODOS MATEMÁTICOS Y DE REPRESENTACIÓN

Hugo Casquero Penelas

Dr. Héctor Gómez Díaz

A Coruña - June 2016

Dedicado a
mi familia, en especial
a mis padres:
Blanca Lourdes Penelas Penelas
Gabriel Casquero Vega

Abstract

Mesh-based immersed approaches shine in a variety of fluid-structure interaction (FSI) applications such as, e.g., simulations where the solid undergoes large displacements or rotations, particulate flow problems, and scenarios where the topology of the region occupied by the fluid varies in time. In this thesis, a new mesh-based immersed approach is proposed which is based on the use of different types of splines as basis functions. This approach is put forth for modeling and simulating different types of biological cells in blood flow at small scales.

The specific contributions of this thesis are outlined as follows. Firstly, a hybrid variational-collocation immersed technique using nonuniform rational B-splines (NURBS) is presented. Newtonian viscous incompressible fluids and nonlinear hyperelastic incompressible solids are considered. Our formulation boils down to three coupled equations which are the linear momentum balance equation, the mass conservation equation, and the kinematic equation that relates the Lagrangian displacement with the Eulerian velocity. The latter is discretized in strong form using isogeometric collocation and the other two equations are discretized using the variational multiscale (VMS) paradigm. As usual in immersed FSI approaches, we define a background mesh on the whole computational domain and a Lagrangian mesh tailored to the region occupied by each solid. Besides of using NURBS for creating these meshes, the data transfer between the background mesh and the Lagrangian meshes is carried out using NURBS functions in such a way that no interpolation or projection is needed, thus avoiding the errors associated with these procedures. Regarding the time discretization, the generalized-α method is used which leads to a fully-implicit and second-order accurate method. The methodology is validated in two- and three-dimensional settings comparing the terminal velocity of free-falling bulky solids obtained in our simulations with its theoretical value. Secondly, we extend our algorithms in order to use analysis-suitable T-splines (ASTS) as basis functions instead of NURBS. This required to develop isogeometric collocation methods for ASTS which was an open problem. The data transfer between meshes changes significantly from NURBS to ASTS due to the fact that their geometrical mappings are local to patches and elements, respectively. ASTS possess two main advantages with respect to NURBS: (1) ASTS support local h-refinement and (2) ASTS are unstructured. The ASTS-based method is validated solving again the aforementioned benchmark problems and showing the potential of ASTS to decrease the amount of elements needed, thus enhancing the efficiency of the method. Thirdly, capsules, modeled as solid-shell NURBS elements, are proposed as numerical proxies for representing red blood cells (RBCs). The dynamics of capsules are able to reproduce the main motions and shapes observed in experiments with RBCs in both shear and parabolic flows. Hemorheological properties as the Fåhræus and Fåhræus-Lindqvist effects are captured in our simulations. In order to obtain the aforementioned results, it is essential to adequately satisfy the incompressibility constraint close to the fluid-solid interface, which is an arduous task in immersed approaches for fluid-structure interaction. Finally, compound capsules are presented as numerical proxies for cells with nucleus such as, e.g., white blood cells (WBCs) and circulat-

ing tumor cells (CTCs). The dynamics of hyperelastic compound capsules in shear flow are studied in both two- and three-dimensional settings. Moreover, we simulate how CTCs manage to pass through channel narrowings, which is an interesting characteristic of CTCs since it is used in experiments to sort CTCs from blood samples.

Acknowledgments

I am thankful for the people that contributed to the completion of this thesis. In particular, I would like to thank Professor Héctor Gómez Díaz, my advisor, for providing guidance, encouragement, and freedom as I explored the topics presented in this work. Having Hector as advisor was one of the main reasons why I decided to do a Ph.D. in computational mechanics. I also thank Hector for giving me the opportunity of collaborating with other research groups. I spent four months at Carnegie Mellon University working with Professor Yongjie (Jessica) Zhang and her Ph.D. student Lei Liu and one week at University of Pavia working with Professor Alessandro Reali and his postdoc Josef Kiendl. I would like to thank Jessica, Alessandro, Josef and Lei for treating me so well during my stay with them and for all what I have learned from them. I would like to thank Professors Laura De Lorenzis, Ming-Chen Hsu, Ignasi Colominas Ezponda, Simone Morganti, Rafael Alejandro Montenegro Armas, and Ignacio Romero Olleros for their time and service as my committee members. My acknowledgments also go to all the members of the MuSIC group: Carles Bona Casas, Jiangping Xu, Guillermo Vilanova Caicoya, Jesús Bueno Álvarez, Guillermo Lorenzo Gómez, and Adrián Moure Rosende for their discussions that have helped shape the ideas presented in this work. I would like to thank Francisco Javier Bermúdez Cerqueiro for taking care of the maintenance of our department cluster. I would like to thank Alba López Picos for helping with administrative work of our research group. And finally, I am most thankful for the patience and support of my parents, Gabriel Casquero Vega and Blanca Lourdes Penelas Penelas.

This thesis was partially supported by the European Research Council through the FP7 Ideas Starting Grant program (Project #307201) and the Ph.D. student grant UDC-Inditex. Their support is gratefully acknowledged.

Contents

List of Figures

List of Tables

1 Introduction

In the last decades, the development of modeling and simulation methodologies for fluid-structure interaction (FSI) problems has attracted enormous attention in the community of computational mechanics. An accurate, geometrically flexible, robust, and efficient computational method is crucial when it comes to solve challenging FSI problems, which are present in biomedical engineering [15, 16], aerospace engineering [116, 117], civil engineering [86, 118, 98, 45], marine engineering [130, 127], and wind energy [12, 13] to name a few. Up to now, there is no computational FSI method that can be considered the best choice (or even a reasonable choice) for all kinds of FSI problems. Therefore, different types of FSI methods have been developed to deal with FSI applications with distinct characteristics.

Mesh-based FSI methods can be classified in two main groups, namely, boundary-fitted and immersed approaches. In boundary-fitted approaches, an arbitrary Lagrangian-Eulerian (ALE) mesh is tailored to the region occupied by the fluid and a Lagrangian mesh is tailored to the region occupied by the solid, these two meshes usually match at the fluid-solid interface so that the imposition of kinematic conditions can be performed exactly. The ALE mesh needs to be updated in each time step solving a boundary-value problem where the applied Dirichlet conditions are the displacements of the solid at the interface. As the solids moves, the quality of the ALE mesh will decrease and eventually reach a point where remeshing is needed, i.e., all or part of the ALE mesh needs to be re-discretized in space. For a thorough explanation of boundary-fitted methods and application examples to real-world problems, the interested reader is remitted to [17]. In immersed approaches, a background mesh spans the whole computational domain and a Lagrangian mesh is tailored to the region occupied by the solid. Therefore, the need for updating or remeshing a mesh tailored to the fluid region has disappeared. However, we get this advantage at the expense of assuming that the pressure field is continuous at the fluid-solid interface and this is specially troublesome when very thin solids are considered [67].

The motion of the solid is the key factor in order to decide whether to use a boundary-fitted or an immersed approach. If we utilize a boundary-fitted method in an application where the solid undergoes large translations or rotations during the simulation, the ALE mesh will become highly distorted each few time steps, thus compromising the accuracy of the solution. Remeshing can be used to counteract this issue, but, besides of its high computational cost, projections of the velocity and pressure fields are needed between the old and new ALE meshes, which are known to introduce inaccuracies specially when the fluid and solid densities are similar. The aforementioned difficulties are amplified when several solids are considered and these solids move throughout the computational domain, e.g., in particulate flow problems [52]. Moreover, in applications where the topology of the region occupied by the fluid is going to change along the simulation due to contact between different solids, applying a boundary-fitted approach is still an open problem. In all the above-mentioned situations, immersed approaches emerge as a suitable alternative to boundary-fitted methods since the Lagrangian mesh of each solid can move freely on top of the background mesh.

The two main aims of this thesis are to develop mesh-based immersed FSI methods using splines and apply them to cell-scale blood flow, which is one of the particulate flows that has recently gained

more attention due to its connection with different diseases as cancer [77]. Spline blending functions such as, e.g., NURBS [106] and T-splines [113] were brought from the CAD community to the computational mechanics community in [63] and [10], respectively. Their higher-order interelement continuity, pointwise nonnegativity, and geometrical flexibility have led to important advancements in both fluid [2, 11] and solid [38, 81] mechanics. Therefore, the application of splines to mesh-based immersed FSI seems a promising way to improve the accuracy, robustness, and geometrical flexibility of existent mesh-based immersed FSI methods.

This thesis extends the current state of the art in mesh-based immersed FSI methods by presenting the next developments which are outlined as follows:

1. **A NURBS-based immersed approach for fluid-structure interaction**. We present a hybrid variational-collocation, immersed, and fully-implicit formulation for fluid-structure interaction. The technique deals with viscous incompressible fluids and nonlinear hyperelastic solids. NURBS-based isogeometric analysis is employed for the spatial discretization and the generalized-α method is used for time discretization. The data transfer between the background mesh and the Lagrangian meshes is carried out using the NURBS functions as well, which avoids the use of the so-called discretized delta functions often employed in immersed FSI methods. We validate our approach comparing our numerical solutions with theoretical data for free-falling rigid bodies in both two- and three-dimensional settings. See Section 4 and Appendix A for details.

2. **Isogeometric collocation based on analysis-suitable T-splines of arbitrary degree**. In order to extend our immersed FSI approach to T-splines, we had to first develop isogeometric collocation using analysis-suitable T-splines of arbitrary degree as basis functions, which was an open problem. Analysis-suitable T-splines (ASTS) enable to perform local h-refinement while maintaining both the cost of only one point evaluation per degree of freedom and the convergence rates of NURBS-based isogeometric collocation. Moreover, ASTS are able to create highly nonuniform meshes without introducing elements with high aspect ratios, which prevents numerical instabilities that arise in problems with reduced regularity when Neumann boundary conditions are imposed in strong form and elements with high aspect ratio are used. See Section 5 and Appendix B for details.

3. **An ASTS-based immersed approach for fluid-structure interaction**. We expand our mesh-based immersed FSI approach to T-splines and validate our numerical results achieving excellent agreement with theoretical data. T-splines possess two main advantages against NURBS, namely, they support local h-refinement and they are unstructured. These properties have the potential of enhancing the efficiency and flexibility of NURBS-based space discretizations. See Section 6 and Appendix C for details.

4. **Capsules and compound capsules as cell mimics in flow**. We perform simulations in both shear

and parabolic flows with capsules in order to reproduce the main motions and shapes that red blood cells (RBCs) acquire in these flows together with various rheological properties of blood at small scales. Moreover, we present the concept of compound capsule as a numerical proxy of cells with nucleus such as, e.g., white blood cells (WBCs) and circulating tumor cells (CTCs). We perform simulations of hyperelastic compound capsules in shear flow in order to study their dynamics in both two- and three-dimensional settings. Finally, we show the capability of CTCs to go through a channel narrowing, this property is currently used to sort CTCs from blood samples [134]. See Section 7 and Appendix D for details.

The structure of this thesis is as follows. The background information regarding mesh-based immersed approaches for fluid-structure interaction, isogeometric analysis, and microscale blood flow is summarized in Section 2. The research objectives of this thesis are stated in Section 3. Sections 4, 5, 6, and 7 expand on the four contributions mentioned above. The idea of these sections is to describe concisely the goals of each contribution, provide both representative results and conclusions, and refer to the published papers attached as appendices for the details. In each of these sections the most significant contributions are highlighted at the end, as well as future research directions emerging from the work performed. Section 8 reports on collaborations in other research developments that have not been connected with our immersed FSI methods so far, but will be connected with these methods in the near future. Finally, conclusions are drawn in Section 9.

2 Background

2.1 Mesh-based immersed approaches for fluid-structure interaction

In the 1970s, mesh-based immersed methods for fluid-structure interaction were introduced by Peskin [95, 96]. Peskin defined a uniform, Eulerian, background mesh using finite differences. Then, he embedded fiber networks as a simplified way to represent solids that are completely submerged in the fluid. The data transfer between the fiber network and the background mesh was carried out using a discretized Dirac delta function with C^1 continuity. This data transfer is divided in two processes: (1) distributing forces coming from the fiber network in the background mesh and (2) interpolating Lagrangian velocities from Eulerian velocities so as to update the fiber network positions. The background mesh is used to compute the Navier-Stokes equations with a forcing function so as to satisfy linear momentum balance and the fiber network is used to both update the position of the solid and orginally compute the interaction forces before they are distributed to the background mesh using the discretized Delta functions. Peskin's method is called Immersed boundary (IB) method and it is an explicit method in time. The IB method has been used to simulate different problems of biomechanics such as, e.g., prosthetic cardiac valves [93], wave propagation in cochlea [23], biofilm processes [42], and swimming motions of marine warms [46].

In 2004, Liu and his coworkers expanded the IB method by developing a new approach called immersed finite element method (IFEM) [131], which was validated performing simulations of free-falling rigid bodies and comparing the terminal velocity with the theoretical solution. IFEM uses the finite element method [135, 62] for both the representation of the deformable solid and the background mesh. This enables a more reliable representation of bulky solids, taking into account both its volume and its mass. IFEM kept using discretized delta functions for transfering data between meshes, but the discretized delta functions were enhanced so as to handle nonuniform background meshes. Regarding the time integration, IFEM is still an explicit method. IFEM has been applied to solve various problems in biomechanics such as, e.g., stent deployment [51], cell-scale blood flow [91, 90], cell migration and cell-substrate adhesion [89]. However, it is important to note that in various of the aforementioned applications, e.g., in [91, 90], the finite elements are no longer used for the Lagrangian meshes since the authors mention that they are not able to handle the large strains of RBCs, the meshfree reproducing kernel particle method is used for representing the solid instead [88, 87]. In [58], the authors proposed an algorithm inspired in IFEM with two main improvements: (1) a semi-implicit time integration algorithm is proposed and (2) projections between meshes are performed to transfer data between meshes instead of using discretized Dirac delta functions. As in [131], the method is validated solving the same type of benchmark problems involving rigid bulky solids.

In 2015, a new mesh-based immersed approach was derived in [67], which uses the augmented Lagrangian framework developed in [14] as starting point and models the solids using Kirchhoff-Love shell theory. In this method, a fluid and structure subproblems are solved, which are coupled with a combination of Lagrange multipliers and penalty forces in order to weakly enforce the kinematic constraint of velocity continuity at the fluid-solid interface. This method has been validated in various benchmark problems by

comparing the results with those of boundary-fitted approaches and applied in simulating a tri-leaflet bioprosthetic heart valve over the complete cardiac cycle. In combination with boundary-fitted approaches to model artery walls, this approach have led to one of the most realistic in silico simulations of heart valves [60, 61].

2.2 Isogeometric analysis

Isogeometric analysis (IGA) was introduced in [63] and thoroughly explained in [37]. IGA was presented with the aim of reusing a computer aided design (CAD) model directly in finite element analysis (FEA), thus simplifying the tedious communication process between design and analysis. One of the main steps in this communication process is to build an analysis-suitable mesh from the CAD geometry. For real-world designs, building this mesh usually needs a high number of manual interactions and the final mesh is just an approximation of the geometry embedded in the CAD file, which may lead to important errors in the analysis process. The above-explained issue is mostly due to the fact that CAD and FEA programs use different types of basis functions in order to represent the geometries; namely, nonuniform rational B-splines (NURBS) [106, 97] and Lagrange polynomials [135, 62] are the most widespread basis functions in CAD and FEA, respectively. IGA proposes to use basis functions coming from CAD to both represent the geometry and the solution fields in FEA using the isoparametric concept. This idea brings closer a potential seamless integration of CAD and FEA, but a second challenge needs to be addressed. Up to now, CAD programs represent objects through surface parameterization instead of volume parameterization. This is an important issue since most applications in analysis require volume parameterizations. Therefore, algorithms to create analysis-suitable spline parameterizations of volumes from surface spline parameterizations in a automatic way are needed. To the best of our knowledge, this is still an open problem, but initial work in this direction has been carried out in [132, 133, 128, 84, 85]. However, there are certain fields in FEA that work with either curve or surface representations such as, e.g., beams and shells. For these fields, several novel spline-based computational methods have been developed in the last years [104, 72, 74, 73, 20, 21, 75, 33].

A key difference between Lagrange polynomials and NURBS is that Lagrange polynomials have C^0 inter-element continuity for all degrees, but NURBS are able to obtain up to C^{p-1} inter-element continuity where p is the degree of the approximation space. This higher inter-element continuity of NURBS have led to important benefits in a large variety of analysis fields where CAD programs are not necessarily used such as, e.g., fourth-order and sixth-order partial differential equations can be solved in primal form [54, 27, 124, 41, 82], the strong form of partial differential equations can be collocated in a straightforward manner [4, 108, 3, 56, 71, 40, 72, 55], enhanced robustness in solid mechanics can be achieved compared to Lagrange polynomials [81], improved accuracy per degree of freedom in fluid mehcanics [2], among others.

The major limitation of NURBS is that they do not possess local refinement capabilities, thus compromising the efficiency of NURBS-based computational methods. In order to overcome this limitation,

T-splines, a generalization of NURBS presented in CAD in [113], where brought to FEA in [10]. T-splines support local h-refinement, but its original definition could lead to blending functions linearly dependent [28] and polynomial partition of unity was not guaranteed. In [80], the subset of analysis-suitable T-splines (ASTS) was defined for cubics and extended to arbitrary degree in [19]. ASTS enable to perform local h-refinement while maintaining all the important mathematical and geometrical properties of NURBS. ASTS have recently been applied to various fields in computational mechanics such as, e.g., dynamic brittle fracture [24], cohesive zone modeling [121], gradient damage models [122], large deformation frictionless contact [43], acoustics [114], and boundary-fitted fluid-structure interaction [14].

2.3 Cell-scale blood flow

The main constituents of blood are:

- Blood Plasma. The blood plasma is an aqueous solution containing numerous chemical species, from ions, mainly Na^+, K^+, Ca^{2+}, and Cl^-, to macromolecules, ranging up to 500-kilodalton molecular weight. Mechanically, it behaves as a Newtonian incompressible fluid whose average density and dynamic viscosity are $1.025\,\mathrm{g/cm^3}$ and $1.2\,\mathrm{mPa\,s}$, respectively [70].

- Red blood cells (RBCs). RBCs carry oxygen from the lungs to the tissues and carbon dioxide from the tissues to the lungs. In mammals, RBCs are non-nucleated cells, namely, they are composed by a lipid bilayer membrane with an attached spectrin network called cytoskeleton which encapsulates a concentrated hemoglobin solution. In a quiescent fluid, healthy RBCs exhibit biconcave shapes. With respect to its mechanical behavior, the lipid bilayer and the cytoskeleton give bending and shear resistance to the cells, respectively. The concentrated hemoglobin solution behaves as a Newtonian incompressible fluid. Under physiological conditions, both the RBC volume and external area are nearly constant. In arteries and veins, the concentration of RBCs acquires a nearly constant value which average is 45%. However, in microcirculation, the concentration of RBCs varies strongly between different vessels due to the so-called plasma skimming effect [78].

- White blood cells (WBCs). WBCs are part of the immune system. WBCs are formed by a membrane, a nucleus, organelles, and fibers suspended in a viscous cytosol [119]. There are various classes of WBCs with different functions and sizes. Regarding its mechanics, the nucleus is significantly stiffer than the rest of the cell. When WBCs are activated, e.g., in the presence of an antigen, their shape and mechanical properties change significantly. In average, WBCs occupy $\sim 1/600$ of total cell volume.

- Platelets. Platelets are non-nucleated disk-shaped cell fragments that are mainly involved in blood coagulation. Platelets possess a membrane and fibers forming a three-dimensional cytoskeleton. Platelets express numerous surface receptors and adhesion molecules that mediate platelet-platelet

interactions as well as interactions with endothelium and plasma components. Activated platelets change their shape, cytoskeletal structure, and mechanical properties [57]. In average, platelets occupy around $1/800$ of total cell volume.

Blood flow in microcirculation is usually treated as a fluid-structure interaction problem since blood cells need to be taken into account explicitly in order to give answer to a large variety of open questions in hemodynamics. Just to give an example, the behavior of an individual red blood cell (RBC) under simple flow conditions such as, e.g., shear or parabolic flows, has been a popular research topic in the last decades in the experimental [48, 9, 100, 125], numerical [6, 103, 102, 39, 99, 94], and analytical [69, 105, 92] communities. RBCs undergo a rich variety of motions in shear flow such as, e.g., the tank-treading (TT) and the tumbling (TU) motions that were firstly observed in experiments in [109, 53]. The reproduction of these motions has been one of the main tests in order to develop computational models of RBCs in flow. RBCs outnumber the rest of blood cells and play a main role in blood rheology. Moreover, they do not get activated by biochemical triggers as WBCs and platelets which simplifies their modeling. The two more widespread continuum-based models of RBCs are vesicles [1] and capsules [101]. Vesicles consider the bending resistance of the lipid bilayer membrane which can be directly controlled through a parameter called bending rigidity, but they neglect the shear resistance of the cytoskeleton. Capsules consider both bending and shear resistances, but both types of resistances are usually controlled through the same parameter, namely, the shear modulus.

The models of other types of blood cells, e.g., white blood cells (WBCs) are less advanced than those for RBCs, but they are receiving more attention in the last years. The main difference between a WBC and a RBC is that a WBC has a nucleus. Recently in [120], a model was presented that aims at taking into account the effect of the nucleus in the dynamics of the cell. A rigid inclusion is considered as the nucleus of the WBC, and the membrane is modeled as a vesicle. This new model is called compound vesicle and the authors start elucidating its dynamics in shear flow performing two-dimensional simulations. In principle, this new model could be applied to other nucleated cells as circulating tumor cells (CTCs). CTCs are cancerous cells that leave the primary tumor, get into the circulatory system through a process called intravasion and are able to live in this new environment. Over time, some of these cells may leave the circulatory system, access a new organ and possibly originate a secondary tumor. These steps are essential in the so-called metastatic cascade. Understanding the dynamics of CTCs in microcirculation could help to develop therapies to stop cancer spreading as well as designing devices that isolate CTCs from a blood sample for diagnosis [77].

3 Research objectives

The objectives of this thesis may be classified in three categories:

- Development and validation of novel numerical methods. The goal is to devise NURBS-based and ASTS-based immersed approaches for fluid-structure interaction. These approaches should take advantage of the higher inter-element continuity of spline basis functions in order to enhance the accuracy and robustness with respect to related mesh-based immersed approaches as the immersed finite element method [131]. Furthermore, the use of spline basis functions would enable to exactly represent geometries that often appear in fluid-structure interaction problems such as, e.g., conic curves and quadric surfaces. The new methods must be validated solving classical benchmark problems with available theoretical solutions.

- Application to biomechanics. The goal is to study the dynamics of different types of blood cells in microscale blood flow. Different authors have reported that Lagrangian polynomials are not able to handle the large deformations of red blood cells in shear and parabolic flows [90, 91], and alternatives approaches as particle-based methods are proposed in instead [88, 87]. The enhanced robustness of solid-shell NURBS elements is invoked to overcome this issue. In order to evaluate the performance of solid-shell NURBS elements, the simulation results should be compared with RBC experiments. Moreover, we aim at capturing the dynamics of cells with nucleus such as, e.g. white blood cells and circulating tumor cells and compare our results with the little available data for this type of cells. Finally, examples in non-trivial geometries should be considered in order to illustrate the geometrical flexibility of our immersed approaches.

- Implementation of the proposed algorithms. Most of the mesh-based immersed approaches for fluid-structure interaction in the literature have only been implemented in serial codes. In this thesis, the goal is to go one step beyond and perform a parallel implementation of our algorithms.

4 A hybrid variational-collocation immersed approach for fluid-structure interaction using NURBS

Let us begin summarizing our immersed formulation for a FSI problem at the continuous level. Let $d = \{2, 3\}$ be the number of spatial dimensions. Let Ω_t^f and $\Omega_t^s \subset \mathbb{R}^d$ be two open sets that represent the time-dependent domains occupied by a viscous incompressible fluid and a hyperelastic incompressible solid, respectively, with Γ_t^f and Γ_t^s to be their corresponding boundaries. These two regions meet at the fluid-solid interface Γ_t^I ($\Gamma_t^I = \overline{\Omega_t^f} \cap \overline{\Omega_t^s}$). Let $\Omega \subset \mathbb{R}^d$ be an open set that represents the domain occupied by both the fluid and the solid, that is, $\overline{\Omega} = \overline{\Omega_t^f \cup \Omega_t^s}$, we assume Ω to be time-independent for both simplicity considerations and since it holds true in all the simulations presented in this thesis. Let Γ be the boundary of Ω. Let $\boldsymbol{v} : \Omega \times (0, T) \mapsto \mathbb{R}^d$, $p : \Omega \times (0, T) \mapsto \mathbb{R}$, and $\boldsymbol{u} : \Omega_0^s \times (0, T) \mapsto \mathbb{R}^d$ denote the Eulerian velocity, the Eulerian pressure, and the Lagrangian displacement, respectively, where $(0, T)$ is the time interval of interest and Ω_0^s is a reference configuration of Ω_t^s. Let $\boldsymbol{X} \in \Omega_0^s$, $\boldsymbol{x} \in \Omega$, and $\boldsymbol{\varphi} : \Omega_0^s \times (0, T) \mapsto \Omega_t^s$ be a material point, a spatial point, and the mapping $\boldsymbol{\varphi}(\boldsymbol{X}, t) = \boldsymbol{X} + \boldsymbol{u}(\boldsymbol{X}, t)$, respectively.

The three equations that define our problem are the linear momentum balance equation, the mass conservation equation, and the kinematic relation between the Eulerian velocity and the Lagrangian displacement, i.e.,

$$\rho^f \left(\frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v} \cdot \nabla_{\boldsymbol{x}} \boldsymbol{v} \right) = \nabla_{\boldsymbol{x}} \cdot \boldsymbol{\sigma}^f + \rho^f \boldsymbol{g} + \boldsymbol{\mathcal{F}} \qquad \text{in} \quad \Omega \times (0, T) \,, \tag{1}$$

$$\nabla_{\boldsymbol{x}} \cdot \boldsymbol{v} = 0 \qquad \text{in} \quad \Omega \times (0, T) \,, \tag{2}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} = \boldsymbol{v} \qquad \text{in} \quad \Omega_t^s \times (0, T) \,, \tag{3}$$

with

$$\boldsymbol{\mathcal{F}} = \begin{cases} 0, & \boldsymbol{x} \in \Omega_t^f \\ \left(\rho^f - \rho^s\right)\left(\frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v} \cdot \nabla_{\boldsymbol{x}} \boldsymbol{v} - \boldsymbol{g}\right) + \nabla_{\boldsymbol{x}} \cdot \left(\boldsymbol{\sigma}^s - \boldsymbol{\sigma}^f\right), & \boldsymbol{x} \in \Omega_t^s \end{cases} , \tag{4}$$

$$\boldsymbol{\sigma}^f = -p\boldsymbol{I} + 2\mu \nabla_{\boldsymbol{x}}^{\text{sym}} \boldsymbol{v} \,, \tag{5}$$

$$\boldsymbol{\sigma}^s = -p\boldsymbol{I} + \boldsymbol{F}\boldsymbol{S}\boldsymbol{F}^T / J \,, \tag{6}$$

where ρ^f and ρ^s are the fluid and solid densities, respectively, $\boldsymbol{\sigma}^f$ and $\boldsymbol{\sigma}^s$ are the fluid and solid Cauchy stress tensors, respectively, $\boldsymbol{g}$ is an external body force acting on the system, $\boldsymbol{I}$ denotes the identity tensor in $\mathbb{R}^{d \times d}$, μ is the dynamic viscosity of the fluid, $\nabla_{\boldsymbol{x}}^{\text{sym}}(\cdot)$ is the symmetric gradient operator given by $\nabla_{\boldsymbol{x}}^{\text{sym}} \boldsymbol{v} = (\nabla_{\boldsymbol{x}} \boldsymbol{v} + \nabla_{\boldsymbol{x}} \boldsymbol{v}^T)/2$, $\boldsymbol{F} : \Omega_0^s \times (0, T) \mapsto \mathbb{R}^{d \times d}$ is the deformation gradient given by $\boldsymbol{F} = \nabla_{\boldsymbol{X}} \boldsymbol{\varphi} = \boldsymbol{I} + \nabla_{\boldsymbol{X}} \boldsymbol{u}$, $\boldsymbol{S}$ is the second Piola-Kirchhoff stress tensor of the considered hyperelastic incompressible material, and $J = \det(\boldsymbol{F})$ is the Jacobian determinant which is exactly 1 at the continuous level since the solid is incompressible.

We must impose suitable boundary and initial conditions to the above equations in order to obtain a well-posed FSI problem. In order to impose boundary conditions, we divide the boundary Γ in two non-overlapping sets $\mathcal{B}^s = \Gamma \cap \Gamma_t^s$ and $\mathcal{B}^f = \Gamma \cap \Gamma_t^f$, which verify that $\Gamma = \overline{\mathcal{B}^s \cup \mathcal{B}^f}$ and $\emptyset = \mathcal{B}^s \cap \mathcal{B}^f$. It is

important to note that $\mathcal{B}^s \neq \Gamma_t^s$ since at least part of the solid is embedded in the fluid. On the interface Γ_t^I, two boundary conditions are imposed, namely, the no-slip boundary condition and the stress compatibility between the fluid and solid Cauchy stress tensors. The no-slip boundary condition is enforce through Eq. (3) and the stress compatibility condition will be enforced in the variational form of (1). On $\mathcal{B}^f$, we assume the standard split in Dirichlet boundary conditions on $\mathcal{B}_D^f$ and Neumann boundary conditions on $\mathcal{B}_N^f$ such that $\mathcal{B}^f = \overline{\mathcal{B}_D^f \cup \mathcal{B}_N^f}$ and $\mathcal{B}_D^f \cap \mathcal{B}_N^f = \emptyset$. On $\mathcal{B}^s$, we consider only homogeneous Dirichlet boundary conditions. Finally, regarding initial conditions, the initial position of the solid, the Lagrangian displacement and the Eulerian velocity must be specified.

Let us continue summarizing our discretization procedure. We first discretized in space using a hybrid variational-collocation technique. The linear momentum balance and mass conservation equations are discretized in weak form (the weak form of these two equations is detailed in Appendix A) using the variational multiscale (VMS) paradigm [64, 65, 11]. However, the kinematic equation between the Eulerian velocity and the Lagrangian displacement is discretized in strong form using isogeometric collocation [4, 32, 108, 56].

The basic idea behind VMS is to split the velocity and the pressure into two components, namely, $\boldsymbol{v} = \widetilde{\boldsymbol{v}} + \boldsymbol{v}'$ and $p = \widetilde{p} + p'$, where $\widetilde{\boldsymbol{v}}$ and $\widetilde{p}$ are finite-dimensional coarse-scale components while $\boldsymbol{v}'$ and p' are infinite-dimensional fine-scale components. Thanks to the stabilized VMS technique that circumvents the Babŭska-Brezzi stability considerations for saddle points problems [5, 26], the same scalar discrete space can be used to represent the weighting and trial functions for $\widetilde{p}$ and each Cartesian component of $\widetilde{\boldsymbol{v}}$. Therefore, let us consider a set of NURBS basis functions $\{N_A^{ED}(\boldsymbol{x})\}_{A=1}^{n_{ED}}$ defined over the physical domain Ω and use this set to define

$$\boldsymbol{v}^h(\boldsymbol{x},t) = \sum_{A=1}^{n_{ED}} \boldsymbol{v}_A(t) N_A^{ED}(\boldsymbol{x}); \quad p^h(\boldsymbol{x},t) = \sum_{A=1}^{n_{ED}} p_A(t) N_A^{ED}(\boldsymbol{x}); \tag{7}$$

$$\boldsymbol{w}^h(\boldsymbol{x}) = \sum_{A=1}^{n_{ED}} \boldsymbol{w}_A N_A^{ED}(\boldsymbol{x}); \quad q^h(\boldsymbol{x}) = \sum_{A=1}^{n_{ED}} q_A N_A^{ED}(\boldsymbol{x}); \tag{8}$$

where $\boldsymbol{w}^h$ and q^h are the weighting functions for the velocity and the pressure, respectively. Following [11], the fine-scale velocity and pressure are expressed in terms of their coarse-scale counterparts as

$$\boldsymbol{v}' = -\tau_M \left(\frac{\partial \boldsymbol{v}^h}{\partial t} + \boldsymbol{v}^h \cdot \nabla_{\boldsymbol{x}} \boldsymbol{v}^h + \nabla_{\boldsymbol{x}} p^h - \nu \Delta_{\boldsymbol{x}} \boldsymbol{v}^h - \boldsymbol{g} \right), \tag{9}$$

$$p' = -\tau_C \nabla_{\boldsymbol{x}} \cdot \boldsymbol{v}^h, \tag{10}$$

with stabilization parameters

$$\tau_M = \left(\frac{4}{\Delta t^2} + \boldsymbol{v}^h \cdot \boldsymbol{G} \boldsymbol{v}^h + C_I \nu^2 \boldsymbol{G} : \boldsymbol{G} \right)^{-\frac{1}{2}}, \tag{11}$$

$$\tau_C = (\tau_M \mathrm{tr}\boldsymbol{G})^{-1}. \tag{12}$$

where $\boldsymbol{G} = [G_{ij}] = \sum_{k=1}^{3} \frac{\partial \xi_k}{\partial x_i} \frac{\partial \xi_k}{\partial x_j}$, $\boldsymbol{x}(\boldsymbol{\xi})$ is the geometrical mapping, $\nu = \mu/\rho^f$ is the kinematic viscosity of the fluid, and C_I is a positive constant equal to 36 in our simulations [37].

Let us now consider a set of NURBS basis functions $\{N_B^{LD}(\boldsymbol{X})\}_{B=1}^{n_{LD}}$ defined over the reference domain Ω_0^s, the scalar discrete space spanned by these basis functions will be used to represent the trial functions for each Cartesian component of the displacement

$$\boldsymbol{u}^h(\boldsymbol{X}, t) = \sum_{B=1}^{n_{LD}} \boldsymbol{u}_B(t) N_B^{LD}(\boldsymbol{X}), \tag{13}$$

With respect to the collocations points used to solve Eq. (3), the set of Greville points $\{\boldsymbol{\tau}_i\}_{i=1}^{n_{LD}}$ defined in [4] is our choice.

We will use the isoparametric concept in order to build both the Eulerian mesh in Ω and the Lagrangian mesh in Ω_0^s. The Lagrangian mesh is pushed forward to physical space using the discrete mapping $\boldsymbol{\varphi}^h(\boldsymbol{X}, t)$.

The final semi-discrete formulation of our variational-collocation immersed method can be stated as: Find $\boldsymbol{v}^h$, p^h, and $\boldsymbol{u}^h$ such that, for all w^h and q^h defined as in Eq. (8) and all $\{\boldsymbol{\tau}_i\}_{i=1}^{n_{LD}}$,

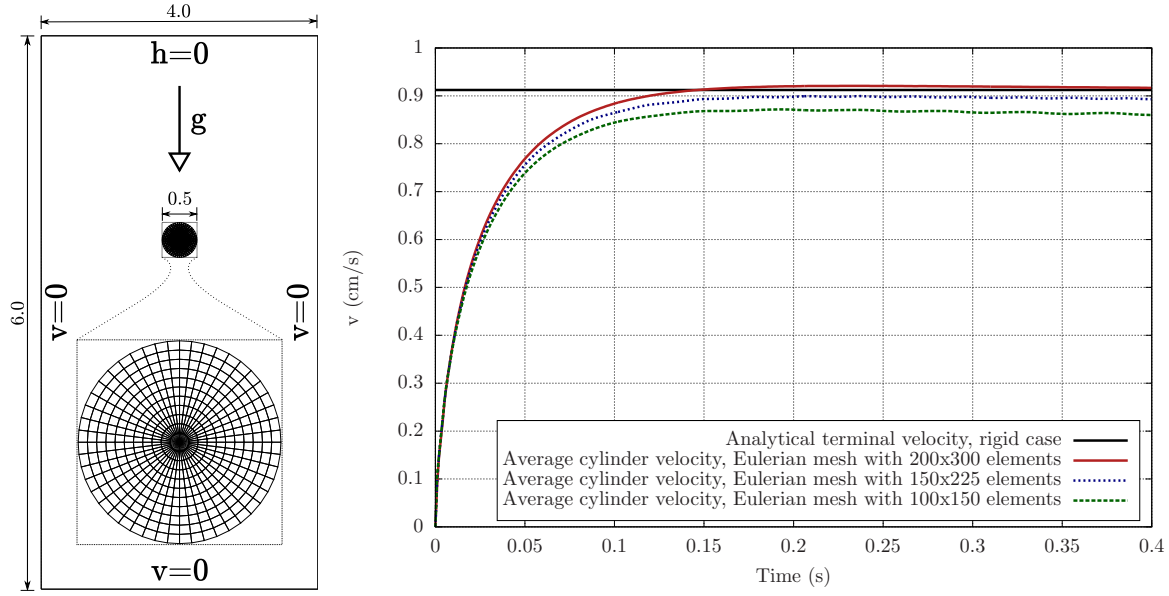
$$B_{MS}^{ED}\left(\{\boldsymbol{w}^h, q^h\}, \{\boldsymbol{v}^h, p^h\}\right) - L^{ED}\left(\boldsymbol{w}^h\right) + B^{LD}\left(\boldsymbol{w}^h, \boldsymbol{v}^h; \boldsymbol{u}^h\right) = 0\,, \tag{14}$$

$$\frac{\partial \boldsymbol{u}^h}{\partial t}(\boldsymbol{\tau}_i, t) = \boldsymbol{v}^h(\boldsymbol{\varphi}^h(\boldsymbol{\tau}_i, t), t)\,, \tag{15}$$

with

$$\begin{aligned} B_{MS}^{ED}\left(\{\boldsymbol{w}^h, q^h\}, \{\boldsymbol{v}^h, p^h\}\right) = &\left(\boldsymbol{w}^h, \frac{\partial \boldsymbol{v}^h}{\partial t} + \boldsymbol{v}^h \cdot \nabla_{\boldsymbol{x}} \boldsymbol{v}^h\right)_\Omega \\ &- \left(\nabla_{\boldsymbol{x}} \cdot \boldsymbol{w}^h, p^h\right)_\Omega + \left(\nabla_{\boldsymbol{x}}^{\mathrm{sym}} \boldsymbol{w}^h, 2\nu \nabla_{\boldsymbol{x}}^{\mathrm{sym}} \boldsymbol{v}^h\right)_\Omega \\ &+ \left(q^h, \nabla_{\boldsymbol{x}} \cdot \boldsymbol{v}^h\right)_\Omega - \left(\boldsymbol{w}^h, \tau_M \boldsymbol{r}_M(\boldsymbol{v}^h, p^h) \cdot \nabla \boldsymbol{v}^h\right)_\Omega \\ &+ \left(\nabla_{\boldsymbol{x}} \boldsymbol{w}^h, (\boldsymbol{v}^h - \tau_M \boldsymbol{r}_M(\boldsymbol{v}^h, p^h)) \otimes \tau_M \boldsymbol{r}_M(\boldsymbol{v}^h, p^h)\right)_\Omega \\ &+ \left(\nabla_{\boldsymbol{x}} \cdot \boldsymbol{w}^h, \tau_C \boldsymbol{r}_C(\boldsymbol{v}^h)\right)_\Omega + \left(\nabla_{\boldsymbol{x}} q^h, \tau_M \boldsymbol{r}_M(\boldsymbol{v}^h, p^h)\right)_\Omega \end{aligned} \tag{16}$$

$$L^{ED}\left(\boldsymbol{w}^h\right) = \left(\boldsymbol{w}^h, \boldsymbol{g}\right)_\Omega + \left(\boldsymbol{w}^h, \boldsymbol{h}/\rho^f\right)_{\mathcal{B}_N^f}\,, \tag{17}$$



(a) Problem setup

(b) Relative area change

Figure 1: *Free falling cylinder. a) Geometrical description of the problem and boundary conditions applied. The inset shows a zoom in of the Lagrangian mesh. The lengths are in centimeters. b) Exact terminal velocity (black line) and time evolution of the average cylinder velocity obtained from the computations (red, blue and green lines). All computations were performed using a Lagrangian mesh composed of* 11×48 *elements. The Eulerian mesh was refined to show convergence of the numerical solution to the exact solution.*

$$
\begin{aligned}
B^{LD}\left(\boldsymbol{w}^h, \boldsymbol{v}^h; \boldsymbol{u}^h\right) = &-\left(\boldsymbol{w}^h, \left(1-\frac{\rho^s}{\rho^f}\right)\frac{\partial \boldsymbol{v}^h}{\partial t}\right)_{\Omega_0^s} \\
&-\left(\boldsymbol{w}^h, \left(1-\frac{\rho^s}{\rho^f}\right)\boldsymbol{v}^h \cdot \nabla_{\boldsymbol{x}} \boldsymbol{v}^h\right)_{\Omega_0^s} \\
&-\left(\nabla_{\boldsymbol{x}}^{\mathrm{sym}} \boldsymbol{w}^h, 2\nu \nabla_{\boldsymbol{x}}^{\mathrm{sym}} \boldsymbol{v}^h\right)_{\Omega_0^s} \\
&+\left(\nabla_{\boldsymbol{x}}^{\mathrm{sym}} \boldsymbol{w}^h, \frac{1}{\rho^f}\boldsymbol{F}\boldsymbol{S}\boldsymbol{F}^T\right)_{\Omega_0^s} \\
&+\left(\boldsymbol{w}^h, \left(1-\frac{\rho^s}{\rho^f}\right)\boldsymbol{g}\right)_{\Omega_0^s},
\end{aligned}
\tag{18}
$$

where $\widetilde{p}$ and p' have been redefined as pressure divided by the fluid density ρ_f as in [11], and $\boldsymbol{h}$ is the prescribed traction vector.

We complete the discretization procedure by applying the Generalized-α method [66] in order to approximate the first order time derivatives that appear in Eqs. (14) and (15) as detailed in Section 3.2 of Appendix A. The implementation algorithm is explained in Section 3.3 of Appendix A.

We start validating the above-explained methodology solving the benchmark problem considered in

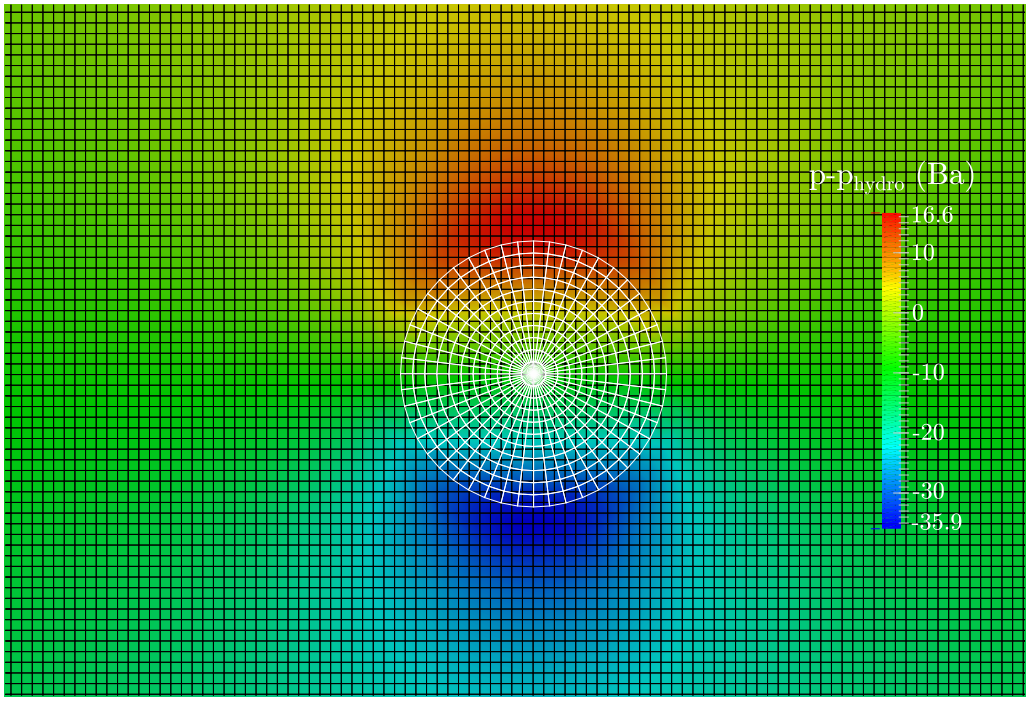


Figure 2: *(Color online) Contour plot of the pressure (subtracting the hydrostatic component) along with the spatial discretizations. The Eulerian mesh is composed of* 200×300 *elements and the Lagrangian mesh has* 11×48 *elements. In all cases, we used quadratic basis functions. Note that the elements of the Lagrangian mesh (white lines) are even bigger than the elements of the Eulerian mesh (black lines), and, yet, the solution is very accurate as shown in Fig. 1 b).*

[58]. In order to perform the numerical simulations included in this Section and in Appendix A, we have developed a serial code on top of the PetIGA framework [36, 22, 123], which adds NURBS discretization capabilities and integration of forms to the scientific library PETSc [8, 7].

The above-mentioned benchmark problem consists of a rigid cylinder of infinite length and radius a which is embedded in a channel of infinite length, infinite height, and width $2L$. The cylinder is denser than the surrounding fluid and is initially at rest. Then, the cylinder will fall due to the action of gravity. The cylinder will increase its velocity until the drag force, which augments with the value of the velocity, plus the buoyancy counterbalance the downward gravity force. After that, the cylinder will keep moving downwards with constant velocity. This velocity is known as terminal or settling velocity (v_T).

Under the assumption of creeping flow, the terminal velocity may be approximated by the expression

$$v_T = \frac{\left(\rho^s - \rho^f\right) g a^2}{4\mu} \left[\ln\left(\frac{L}{a}\right) - 0.9157 + 1.7244 \left(\frac{a}{L}\right)^2 - 1.7302 \left(\frac{a}{L}\right)^4 \right] \tag{19}$$

where g is the Euclidean norm of the gravity acceleration $\boldsymbol{g}$.

When the assumption of creeping flow is reasonable, i.e., when the Reynolds number $R_e = 2a\rho^f v_T/\mu << 1$, Eq. (19) and our numerical solution should be in agreement and as R_e gets close to 1 or higher, they are

expected to give different results since we do not assume creeping flow in our methodology. Moreover, we set a high enough shear modulus in our simulations so that the strains of our cylinder are negligible. We have used the next second Piola-Kirchhoff stress tensor

$$\boldsymbol{S} = \mu^s J^{-2/d}\left(\boldsymbol{I} - \frac{1}{d}\text{tr}(\boldsymbol{C})\boldsymbol{C}^{-1}\right) + \frac{1}{2}\kappa^s\left(J^2 - 1\right)\boldsymbol{C}^{-1} \tag{20}$$

where μ^s is the shear modulus of the considered solid and $\boldsymbol{C} = \boldsymbol{F}\boldsymbol{F}^T$ is the Cauchy-Green deformation tensor. Note that following [131], we have included the compressible part of the stored energy functional to derive $\boldsymbol{S}$. Consistently with what was reported in [131], we found that disregarding the dilatational part of the stored energy function produced non-negligible errors in the incompressibility constraint close to the fluid-solid interface. Therefore, following [131] we consider the last term of Eq. (20) as a penalty term in order to keep J close enough to one near the interface. In this context, κ^s should be interpreted as a penalty parameter to be adjusted in simulations (we have used values of κ^s on the order of magnitude of $\mu^s/10$)

Our two-dimensional setting and the boundary conditions applied are drawn in Fig. 1 a). The physical parameters that intervene in this benchmark problem are: gravity acceleration $g = 981$ cm/s^2, dynamic viscosity $\mu = 5$ dyn/cm^2, fluid density $\rho^f = 1$ g/cm^3, solid density $\rho^s = 1.25$ g/cm^3, and shear modulus $\mu^s = 33550$ g/(cm s^2). Regarding the spatial discretization, we use quadratic NURBS elements for both the Eulerian an Lagrangian meshes. Regarding the time discretization, the time step was set to $\Delta t = 0.001$ s. Fig. 1 b) shows how we get closer to the theoretical value of v_T as we refine the Eulerian mesh. Fig. 2 plots the pressure field minus its hydrostatic component close to the solid. In this case, the pressure does not undergo sudden variations close to the fluid-solid interface. This suggests that the assumption of continuous pressure at the fluid-solid interface, which is inevitable in immersed approaches, is not going to have a significant impact on the accuracy of the solution. However, this changes when thin solids are considered as can be seen in the last example shown in Appendix A. We will address this issue in more detail in Section 7 and Appendix D. Fig. 3 shows the two vortices that appear close to the cylinder. We validate the methodology in a three-dimensional setting and study the robustness of our algorithm in Sections 4.2 and 4.3 of Appendix A, respectively.

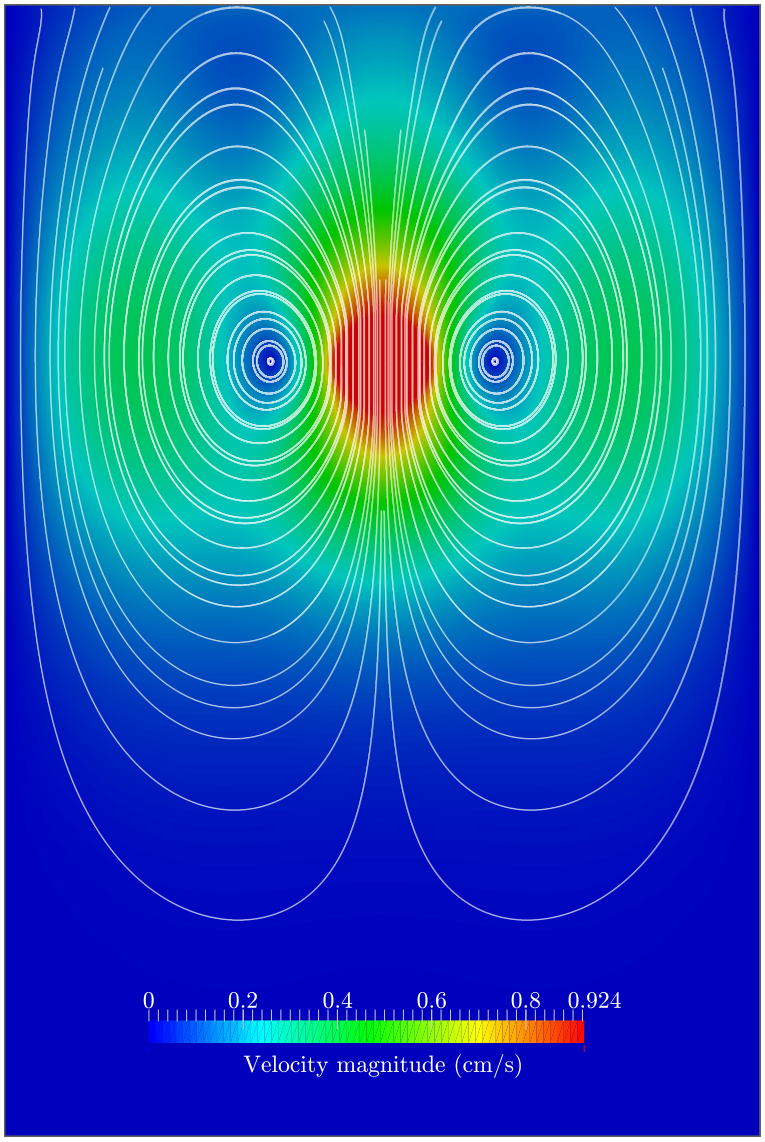


Figure 3: *(Color online) Velocity magnitude along with the streamlines at time* $t = 0.35$ s *for the free-falling cylinder example. The plot clearly shows two large vortices close to the cylinder, which is the expected laminar-flow solution. The Eulerian mesh is composed of* 200×300 *elements and the Lagrangian mesh has* 11×48 *elements. At the scale of the plot, the solution looks symmetric as expected.*

Contributions

The most significant contributions of this work, which has been published as a research paper in [29], are:

1. A novel hybrid variational-collocation immersed approach using NURBS is presented and validated using classical benchmark problems.
2. The formulation is fully-implicit which implies updating the positions of the solids in each nonlinear iteration of each time step.
3. The data transfer between meshes is carried out without using interpolation or projection procedures, thus avoiding the errors associated with these procedures.
4. The higher inter-element continuity of NURBS functions enables to handle larger distortions of the Lagrangian meshes in our immersed approach, thus enhancing the robustness of the algorithm. These results are consistent with previous works in structural mechanics [81].

Open lines for research

The main open ideas derived from this work are:

1. Generalizing the algorithms in order to use splines spaces with local h-refinement capabilities such as, e.g., ASTS [19] and hierarchical B-splines [126, 107].
2. Studying in detail the implications of the poor approximation properties of the pressure field in immersed approaches for fluid-structure interaction.
3. Developing a MPI implementation of the proposed algorithms, which is a non-standard problem since the Lagragian meshes move freely on top of the background mesh and data transfer is needed between these meshes.
4. Applying the methodology to cell-scale blood flow.

5 Analysis-suitable T-splines as a basis for isogeometric collocation

In [113], T-splines were introduced in the CAD community as a generalization of NURBS. T-splines were developed to overcome the main drawbacks of NURBS in design such as, for example, the high amount of superfluous control points needed to represent complex geometries [111] and the difficulty to create watertight surfaces in general cases [112]. The mathematical study of T-splines has been an active field in the last years. As a consequence of this study, the subset of ASTS was defined, first for cubics [80, 110, 79] and then for arbitrary degree [19, 18, 25]. ASTS are a subset of T-splines that satisfies the following important mathematical properties:

(1) Linear independence of the blending functions.

(2) Polynomial partition of unity of the blending functions.

(3) Local h-refinement is available.

(4) Optimal convergence rates in analysis under h-refinement.

(5) Each blending function is pointwise non-negativity.

(6) Affine covariance.

(7) Convex hull property.

(8) Dual compatibility [18].

(9) Forward and backward compatible with NURBS.

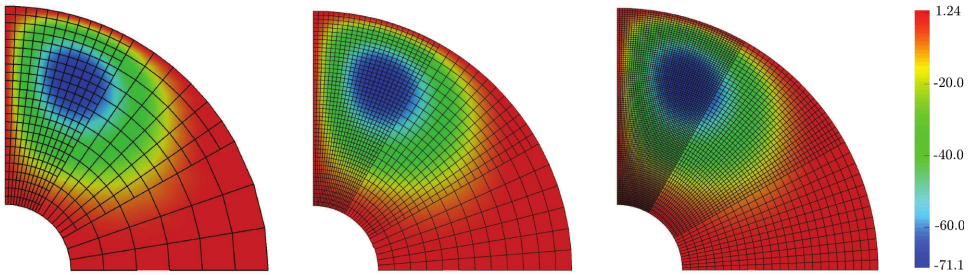
In other words, ASTS of arbitrary degree maintain all the important mathematical and geometrical properties of NURBS and possess local h-refinement capabilities.

The construction of ASTS of arbitrary degree and our algorithms in order to collocate both second-order and fourth-order PDEs in strong form are detailed in Appendix B. In these algorithms, Greville points are used as collocation points.

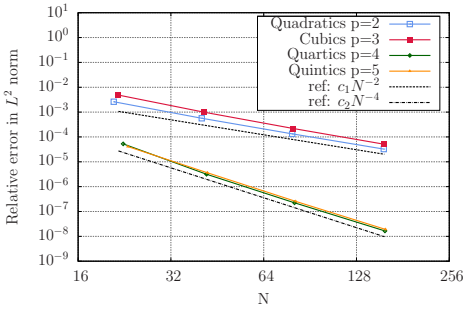
In Section 4 of Appendix B, we start analyzing the convergence rates of the proposed algorithms. For that, second-order elliptic problems with smooth and rough solution fields are considered. Firstly, we consider a reaction-diffusion boundary-value problem

$$-\Delta u + u = f \qquad \text{in} \quad \Omega, \tag{21}$$

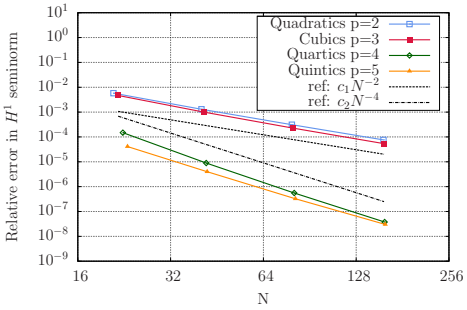
$$u = 0 \qquad \text{on} \quad \Gamma. \tag{22}$$



(a) Problem setup



(b) L^2 norm



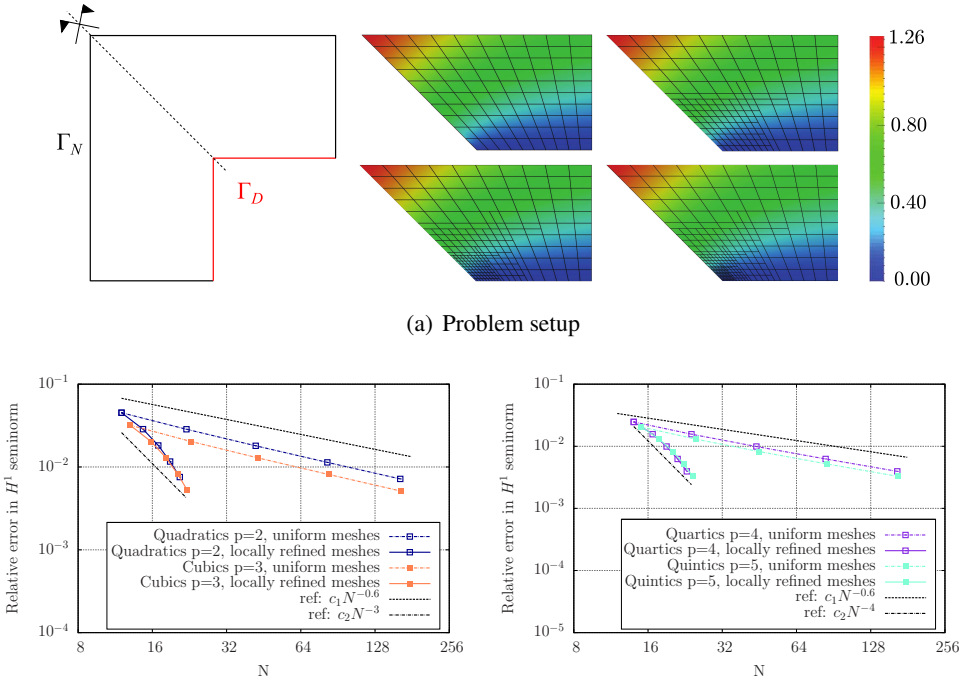
(c) H^1 seminorm

Figure 4: *a) Second-order problem with smooth solution. Numerical solution on three different meshes using $p = 4$. b) L^2 norm of the relative error between the numerical and analytical solutions. c) H^1 seminorm of the relative error between the numerical and analytical solutions.*

The computational domain Ω is a quarter of annulus situated in the positive quadrant of the Cartesian coordinate system. The inner radius R_i is equal to 1.0 and the outer radius R_o is equal to 4.0. As in [108], we manufacture the source term f such that the exact solution to the problem is

$$u(x, y) = \Theta(x, y)^2[\varrho(x, y)^2 - 1][\varrho(x, y)^2 - 16]\sin(x), \tag{23}$$

where $\varrho(x, y) = \sqrt{x^2 + y^2}$ and $\Theta(x, y) = \arctan(y/x)$ are the polar coordinates. We solve this problem using quadratic, cubic, quartic, and quintic ASTS. We start considering a NURBS patch with 6×9 elements that represents the domain Ω exactly. Then, we perform quad-tree refinement using three levels of refinement in each mesh as it is illustrated in Fig. 4 a). We choose this type of h-refinement to be able to compare with the data in [108]. Fig. 4 a) also shows the numerical solutions for various quartic ASTS meshes. Fig. 4 b) and c) plot the convergence rates of the L^2 norm and the H^1 seminorm of the relative error between the numerical and analytical solutions, respectively. The plots show the evolution of the errors with respect to the square root of the total number of degrees of freedom (N). The data suggest that the rates of convergence are of the type $O(p)$ for even orders and $O(p-1)$ for odd degrees, where p is the order of our ASTS discrete space. These convergence rates are consistent with those obtained with both NURBS [4] and hierarchical NURBS [108] when Greville points are used as collocation points.



(a) Problem setup

(b) L^2 norm

(c) H^1 seminorm

Figure 5: *a) Second-order problem with rough solution. Numerical solution on three different meshes using $p = 2$. b) H^1 seminorm of the relative error between the numerical and analytical solutions using globally and locally h-refined quadratic and cubic meshes. c) H^1 seminorm of the relative error between the numerical and analytical solutions using globally and locally h-refined quartic and quintic meshes.*

Secondly, we consider a problem governed by the Laplace equation on a L-shaped domain

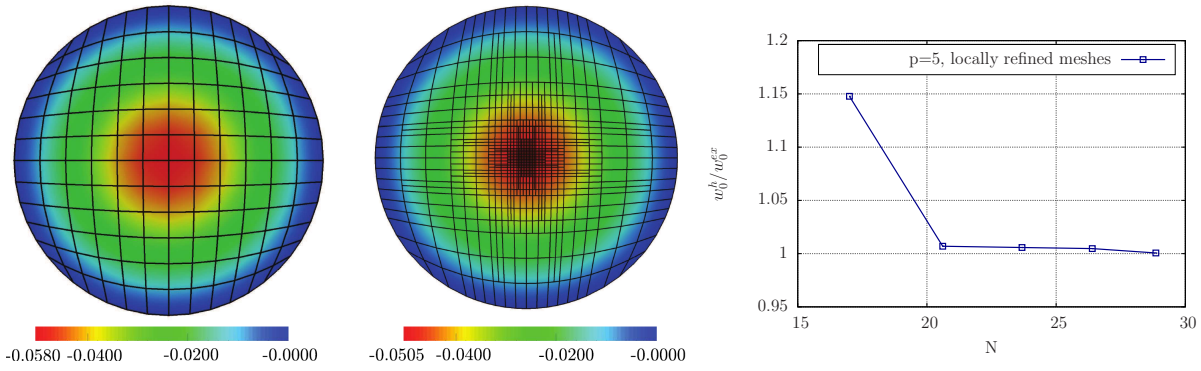
$$\Delta u = 0 \qquad \text{in} \quad \Omega, \tag{24}$$

$$u = u_\Gamma \qquad \text{on} \quad \Gamma_D, \tag{25}$$

$$\nabla u \cdot \boldsymbol{n} = h_\Gamma \qquad \text{on} \quad \Gamma_N. \tag{26}$$

where $\Omega = [-1, 1]^2 \setminus ([0, 1] \times [-1, 0])$, Γ_D is composed by the re-entrant boundaries, and $\Gamma_N = \Gamma \setminus \Gamma_D$ [see Fig. 5 a)]. A solution is manufactured so as to create a singularity in the re-entrant corner. As expected, this singularity harms the convergence rates when uniform h-refinement is utilized as can be seen in Fig. 5 b) and c). Nevertheless, if we perform local h-refinement as shown in Fig. 5 a), we are able to significantly improve the convergence rates as shown in Fig. 5 b) and c), namely, $O(0.6)$ is approximately the convergence rate for all p. In Appendix B, another two second-order problems characterized by the presence of sharp layers and spikes are solved.

Finally, we consider a simply supported Kirchhoff plate with a point load applied in its center, which mechanical response is governed by a fourth-order partial differential equation. The radius of the plate is



(a) Deflection field (b) Convergence study

Figure 6: *a) Initial NURBS mesh (left) and final quintic ASTS mesh with five levels of refinement (right) along with the corresponding computed deflection fields. Quintics meshes are used. b) Convergence of the numerical deflection at the center of the plate using quintic ASTS. We start with a uniform NURBS mesh and introduce a level of refinement in each new mesh. N denotes the square root of the total number of degrees of freedom.*

$r = 1$ and the point load applied is $P = 1$. The PDE and boundary conditions that we must collocate in this case are the following

$$\Delta^2 w = \Upsilon \quad \text{in} \quad \Omega, \tag{27}$$

$$w = 0 \quad \text{on} \quad \Gamma, \tag{28}$$

$$\vartheta \Delta w + (1 - \vartheta)\boldsymbol{n} \cdot (\nabla\nabla w)\boldsymbol{n} = 0 \quad \text{on} \quad \Gamma. \tag{29}$$

where w is the deflection of the plate, ϑ is the Poisson ratio assumed to be 0.3, $\boldsymbol{n}$ is the unit outward normal to the plate, and Υ is load per unit surface equivalent to the point load P whose discretization is explained in detail in Appendix B.

We reproduce the geometry of the plate exactly using quintic ASTS meshes. We start with a uniform NURBS mesh and perform local h-refinement in several levels in order to enhance the resolution near the center of the plate. Fig. 6 a) shows the coarsest and finest meshes along with the corresponding computed deflection fields. For each mesh, we compute the deflection under the load w_0^h and study how it gets closer to its known exact value w_0^{ex} as we introduce more refinement levels as can be seen in 6 b), namely, the relative error of the numerical solution with respect to the exact solution for the mesh with five levels of refinement is 0.06%.

Contributions

The most significant contributions of this work, which has been published as a research paper in [32], are:

1. Isogeometric collocation methods for analysis-suitable T-splines of arbitrary degree are presented, which use Greville points as collocation points.

2. The proposed algorithms maintain the cost of only one point evaluation per degree of freedom typical of standard NURBS-based collocation and enable to perform local h-refinement, thus enhancing the efficiency of the method. Note that other spline spaces with local h-refinement capabilities, e.g., hierarchical NURBS are not able to maintain the cost of only one point evaluation per degree of freedom when they are used as a basis for isogeometric collocation.

3. ASTS are able to easily make highly nonuniform meshes without having elements with high aspect ratios. This is particularly interesting in isogeometric collocation since instabilities may arise when Neumann boundary conditions are imposed in strong form and elements with high aspect ratio are present, this situation can be easily avoid with the use of ASTS.

Open lines for research

The main open ideas derived from this work are:

1. The algorithms of this work could be used for a different set of collocation points instead of Greville points. Recently, the variational collocation paradigm was introduced using NURBS as a basis in [55]. There, a new set of collocation points is defined, the so-called Cauchy-Galerkin points, which outperform Greville points for odd degrees raising the convergence rate to p instead of $p - 1$. The generalization of Cauchy-Galerkin points to spline spaces with local h-refinement capabilities seems a promising idea.

2. Kirchhoff-Love shells is one of the applications that has benefited more from the arrival of isogeometric analysis. However, the current formulations of isogeometric Kirchhoff-Love shells use the Galerkin method to discretize the fourth-order PDE that governs this problem. Developing collocation methods for such a PDE is an appealing direction for future research and ASTS seem the proper basis functions to use due to its important geometrical properties for creating surface representations and its local h-refinement capabilities to enhance the efficiency in analysis.

3. Developing analysis-suitable volumetric T-splines in order to solve problems in three-dimensional settings using isogeometric collocation.

6 An ASTS-based immersed approach for fluid-structure interaction

In this Section and in Appendix B, we extend our immersed algorithms in order to handle ASTS and show its superior efficiency due to their local h-refinement capabilities. Let us start emphasizing on a topic quite specific of immersed approaches for fluid-structure interaction, which in addition varies substantially for NURBS-based and ASTS-based techniques. In order to do so, let us repeat here Eqs. (14) and (15) for completeness:

$$B_{MS}^{ED}\left(\{\boldsymbol{w}^h, q^h\}, \{\boldsymbol{v}^h, p^h\}\right) - L^{ED}\left(\boldsymbol{w}^h\right) + B^{LD}\left(\boldsymbol{w}^h, \boldsymbol{v}^h; \boldsymbol{u}^h\right) = 0\,, \tag{30}$$

$$\frac{\partial \boldsymbol{u}^h}{\partial t}(\boldsymbol{\tau}_i, t) = \boldsymbol{v}^h(\boldsymbol{\varphi}^h(\boldsymbol{\tau}_i, t), t)\,, \tag{31}$$

with

$$\begin{aligned} B_{MS}^{ED}\left(\{\boldsymbol{w}^h, q^h\}, \{\boldsymbol{v}^h, p^h\}\right) = &\left(\boldsymbol{w}^h, \frac{\partial \boldsymbol{v}^h}{\partial t} + \boldsymbol{v}^h \cdot \nabla_{\boldsymbol{x}} \boldsymbol{v}^h\right)_\Omega \\ &- \left(\nabla_{\boldsymbol{x}} \cdot \boldsymbol{w}^h, p^h\right)_\Omega + \left(\nabla_{\boldsymbol{x}}^{\mathrm{sym}} \boldsymbol{w}^h, 2\nu \nabla_{\boldsymbol{x}}^{\mathrm{sym}} \boldsymbol{v}^h\right)_\Omega \\ &+ \left(q^h, \nabla_{\boldsymbol{x}} \cdot \boldsymbol{v}^h\right)_\Omega - \left(\boldsymbol{w}^h, \tau_M \boldsymbol{r}_M(\boldsymbol{v}^h, p^h) \cdot \nabla \boldsymbol{v}^h\right)_\Omega \\ &+ \left(\nabla_{\boldsymbol{x}} \boldsymbol{w}^h, (\boldsymbol{v}^h - \tau_M \boldsymbol{r}_M(\boldsymbol{v}^h, p^h)) \otimes \tau_M \boldsymbol{r}_M(\boldsymbol{v}^h, p^h)\right)_\Omega \\ &+ \left(\nabla_{\boldsymbol{x}} \cdot \boldsymbol{w}^h, \tau_C \boldsymbol{r}_C(\boldsymbol{v}^h)\right)_\Omega + \left(\nabla_{\boldsymbol{x}} q^h, \tau_M \boldsymbol{r}_M(\boldsymbol{v}^h, p^h)\right)_\Omega \end{aligned} \tag{32}$$

$$L^{ED}\left(\boldsymbol{w}^h\right) = \left(\boldsymbol{w}^h, \boldsymbol{g}\right)_\Omega + \left(\boldsymbol{w}^h, \boldsymbol{h}/\rho^f\right)_{\mathcal{B}_N^f}\,, \tag{33}$$

$$\begin{aligned} B^{LD}\left(\boldsymbol{w}^h, \boldsymbol{v}^h; \boldsymbol{u}^h\right) = &-\left(\boldsymbol{w}^h, \left(1 - \frac{\rho^s}{\rho^f}\right) \frac{\partial \boldsymbol{v}^h}{\partial t}\right)_{\Omega_0^s} \\ &- \left(\boldsymbol{w}^h, \left(1 - \frac{\rho^s}{\rho^f}\right) \boldsymbol{v}^h \cdot \nabla_{\boldsymbol{x}} \boldsymbol{v}^h\right)_{\Omega_0^s} \\ &- \left(\nabla_{\boldsymbol{x}}^{\mathrm{sym}} \boldsymbol{w}^h, 2\nu \nabla_{\boldsymbol{x}}^{\mathrm{sym}} \boldsymbol{v}^h\right)_{\Omega_0^s} \\ &+ \left(\nabla_{\boldsymbol{x}}^{\mathrm{sym}} \boldsymbol{w}^h, \frac{1}{\rho^f} \boldsymbol{F}\boldsymbol{S}\boldsymbol{F}^T\right)_{\Omega_0^s} \\ &+ \left(\boldsymbol{w}^h, \left(1 - \frac{\rho^s}{\rho^f}\right) \boldsymbol{g}\right)_{\Omega_0^s}\,, \end{aligned} \tag{34}$$

The computation of the operator B^{LD} in Eq. (30) is not standard. We use Gaussian quadrature rules defined on the parametrization of the Lagrangian mesh. However, the integrands have functions such as, e.g., $\boldsymbol{v}^h$, $\boldsymbol{w}^h$, and their first derivatives, which are defined on the background mesh instead of the Lagrangian mesh. Therefore, for a certain Gauss point with parametric coordinates $\boldsymbol{\xi}_G^{LD}$ in the Lagrangian mesh, we first need to compute its physical location $\boldsymbol{x}_G$ using the geometrical mapping of the Lagrangian mesh, i.e, $\boldsymbol{\varphi}^h$ and then invert the geometrical mapping of the background mesh in order to obtain the parametric coordinates

Table 1: *Background meshes used for the free falling cylinder in a channel.*

Background mesh	Refinement	Smallest element size	Total number of Bézier elements
ASTS1	Three levels of refinement	0.1	1162
NURBS1	Uniform	0.1	3200
ASTS2	Four levels of refinement	0.05	3868
NURBS2	Uniform	0.05	12800
ASTS3	Five levels of refinement	0.025	14428
NURBS3	Uniform	0.025	51200

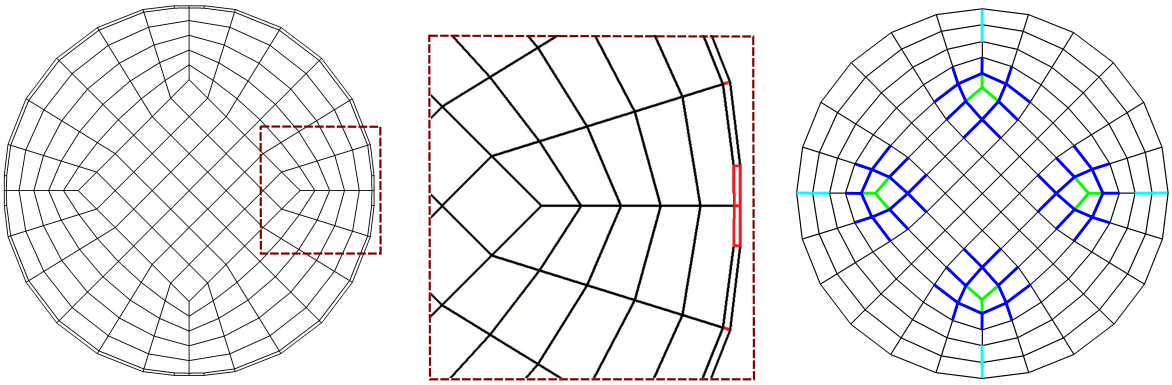
$\boldsymbol{\xi}_G^{ED}$ associated with the physical point $\boldsymbol{x}_G$ in the background mesh. Once we know $\boldsymbol{\xi}_G^{ED}$, we can evaluate functions that are defined on the background mesh using standard procedures of finite elements. Note that an analogous situation occurs when we try to evaluate Eq. (31) at a certain collocation point.

When Bézier extraction is used, which is the case of our ASTS-based implementation, the geometrical mappings are local to elements as in the classical finite element method. Therefore, if we want to compute $\boldsymbol{\xi}_G^{ED}$ from $\boldsymbol{x}_G$, we must first find out the Bézier element in which $\boldsymbol{x}_G$ is located. In other words, we must find $e \in \{1, \dots, N_{el}\}$ and $\boldsymbol{\xi}_G^{ED} \in \square$ such that

$$\boldsymbol{x}_G = \sum_{a=1}^{n^e} \boldsymbol{Q}_a^e N_a^e(\boldsymbol{\xi}_G^{ED}) \,, \tag{35}$$

where N_{el} is the global number of Bézier elements, n^e is the number of ASTS basis functions with support on the element e, N_a^e is the a-th rational ASTS basis function over the element e, and $\boldsymbol{Q}_a^e$ is the a-th control point over the element e. Eq. (35) is a $d \times d$ nonlinear system that we solved using the Newton-Raphson algorithm. It is important to check that the solution $\boldsymbol{\xi}_G^{ED}$ satisfies that $\boldsymbol{\xi}_G^{ED} \in \square$ since Eq. (35) may have roots outside the parent element. Solving Eq. (35) for all Bézier elements would computationally be quite expensive. Fortunately, this is only needed in the first nonlinear iteration of the first time step. In all others nonlinear iterations and time steps, we proceed as follows. For a certain point $\boldsymbol{\xi}_G^{LD}$, we store the Bézier element in which its physical counterpart fell at the previous nonlinear iteration. When we need to find the Bézier element in which it falls in the current nonlinear iteration, we first try only in the Bézier element that hosted the point in the previous nonlinear iteration and if that element does not find the parametric point $\boldsymbol{\xi}_G^{ED}$, we try with the neighbor elements. In our NURBS-based implementation, we do not use Bézier extraction, i.e., we work directly with NURBS patches. This simplifies to invert the geometrical mapping of the background mesh since this mapping is no longer local to elements, but patches. Besides of that, various common NURBS geometrical mappings such as, e.g., rectangles and parallelograms can be inverted analytically which avoids solving Eq. (35) using the Newton-Raphson algorithm.

The above-explained integration process used for B^{LD} and L^{LD} is suboptimal since the functions that



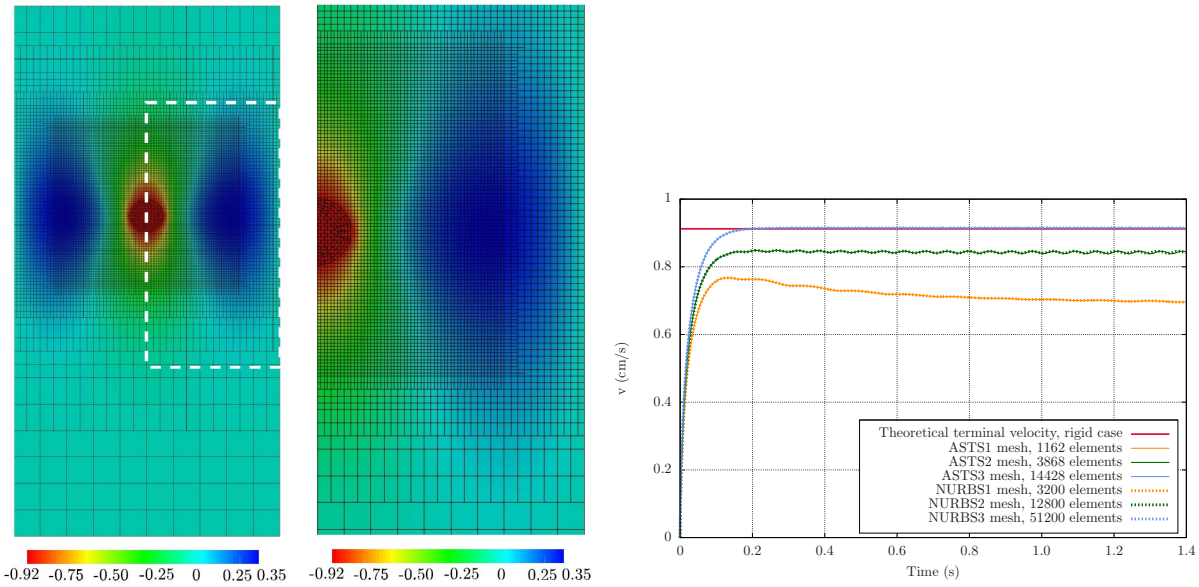
(a) Unstructured T-mesh (b) Detail of the T-mesh (c) Bézier mesh

Figure 7: *(Color online) T-mesh and Bézier mesh of a circle. (a) T-mesh with four extraordinary points which avoids the presence of degenerated points in the parameterization. (b) Zoom-in of the T-mesh close to one of the extraordinary points. The color code in the edges of the T-mesh defines the knot interval configuration. All edges have a knot interval of 1 associated, except those colored in red which have associated a zero-length knot interval. (c) Bézier mesh in physical space. The basis functions are C^∞ everywhere, except at the boundaries of the Bézier elements, where the continuity is reduced. The continuity across black Bézier-element boundaries is C^2. Dark blue lines represent element boundaries of C^1 continuity due to the presence of extraordinary points. Green and light-blue lines represent element boundaries of C^0 continuity due to the presence of extraordinary points and our choice to obtain exact geometry, respectively.*

are defined on the background mesh may have lines of reduced continuity in the interior of the integration regions which are the elements of the Lagrangian mesh. However, the higher inter-element continuity of spline functions is used in order to alleviate this issue, e.g., an Eulerian mesh constructed with cubic splines is needed in order to avoid having functions with C^0-continuous lines in the interior of the integration regions.

We validate our ASTS-based technique solving a benchmark problem that we solved in Section 4 and Appendix A using NURBS, namely, the free falling cylinder in a channel. For performing all the simulations involving ASTS of this Section and Appendix C, a serial code was directly developed on top of the scientific library PETSc [8, 7] without using PetIGA [36]. All the data structures used in PetIGA, as the so-called DA's PETSc structures, rely heavily on the assumption that the mesh has a tensor product structure, this is no longer the case in ASTS and hence PetIGA is not suitable for ASTS-based isogeometric analysis.

ASTS and NURBS, unlike Lagrange piecewise polynomials, can represent a circle exactly. Moreover, the unstructured nature of ASTS enables to exactly represent the circle without the need to introduce singular points in the parameterization. We note that this is not possible using single-patch NURBS. The same cubic Lagrangian mesh is used in all simulations which is plotted in Fig. 7 along with its associated T-mesh. Three ASTS Eulerian meshes and three NURBS Eulerian meshes are considered; these meshes are detailed in Table 1. Note that the size of the elements in the uniform NURBS meshes is always picked equal to the size of the smallest elements in a ASTS mesh. This is done to compare local h-refinement against global



(a) Vertical velocity (b) Zoomed-in view (c) Terminal velocity

Figure 8: *(Color online) (a) Eulerian and Lagrangian meshes along with a color plot of the vertical velocity. The Eulerian mesh is locally h-refined where the flow is most complex. T-junctions permitted us to reduce by* 68% *the number of Bézier elements with respect to a uniform mesh that achieves the same accuracy. (b) Zoom in of the inset area in the left panel. (c) Time evolution of the average cylinder velocity for each Eulerian mesh compared against the theoretical terminal velocity (red line). The Lagrangian mesh is the same in all simulations and it is composed of* 156 *Bézier elements and* 209 *control points. Excellent agreement is obtained with respect to the theoretical terminal velocity when enough spatial resolution is used.*

h-refinement. All the Eulerian meshes are C^2-continuous.

Fig. 8 a) draws the ASTS Eulerian mesh with five levels of refinement, the Lagrangian mesh, and the vertical velocity at time $t = 0.75$s. Fig. 8 b) represents a zoom in of the rectangle indicated in Fig. 8 a) in order to show the different levels of refinement. Fig. 8 c) shows how the numerical solution gets closer to the theoretical value of the terminal velocity as we increase the spatial resolution for both ASTS and NURBS meshes. The globally and locally *h*-refined meshes with the same element size give essentially the same solution which shows the potential of ASTS to enhance the efficiency of the algorithm. The relative error for the meshes ASTS3 and NURBS3 is below 0.2%, which seems to be similar to the error in the theoretical solution given the number of digits employed in the constants in Eq. (19). Note that the mesh ASTS3 has 68% less Bézier elements than the mesh NURBS3 which shows the importance of using T-junctions. Finally, we would like to mention that only twenty Bézier elements are behind the solid for the meshes ASTS1 and NURBS1 which is clearly not enough to give a good solution.

In Section 4.2 of Appendix C, we also consider an example involving a partially immersed solid.

Contributions

The most significant contributions of this work, which has been published as a research paper in [31], are:

1. We generalize our algorithms so as to handle ASTS.
2. We discuss how to alleviate the issue of suboptimal quadrature rules that arises in immersed approaches by increasing the inter-element continuity of the background mesh.
3. The use of T-junctions enables to perform local h-refinement where higher spatial resolution is needed, which usually happens close to the fluid-solid interface.
4. The presence of singular points is avoided by introducing extraordinary points.

Open lines for research

The main open ideas derived from this work are:

1. Creating ASTS background meshes that are locally h-refined in an adaptive way as the solid moves. This would highly enhance the efficiency of immersed approaches. Up to now, the adaptivity of ASTS has not been studied and remains and open problem.
2. Developing analysis-suitable volumetric T-splines in order to solve problems in three-dimensional settings.

7 Capsules and compound capsules as cell mimics

In this Section and in Appendix D, we consider capsules as our numerical proxies for RBCs. A capsule is composed by a very thin membrane with a fluid inside. The membrane aims at representing both the bending resistance of the lipid bilayer and the shear resistance of the cytoskeleton. The fluid inside represents the hemoglobin solution which behaves as a Newtonian fluid assumed to have the same viscosity than blood plasma for simplicity.

As mentioned in Sections 1 and 4, immersed approaches consider continuous pressure at the fluid-solid interface. Nevertheless, the exact solution to the FSI problem may have discontinuous pressure at the interface. This leads to discrete pressure spaces with poor approximation properties at the interface. Since the pressure is the Lagrange multiplier responsible for imposing the incompressibility constraint, this may originate a lack of local mass conservation near the interface for practical levels of refinement. This issue becomes particularly problematic when very thin solids are considered, which is the case for all the examples considered in this Section and in Appendix D.

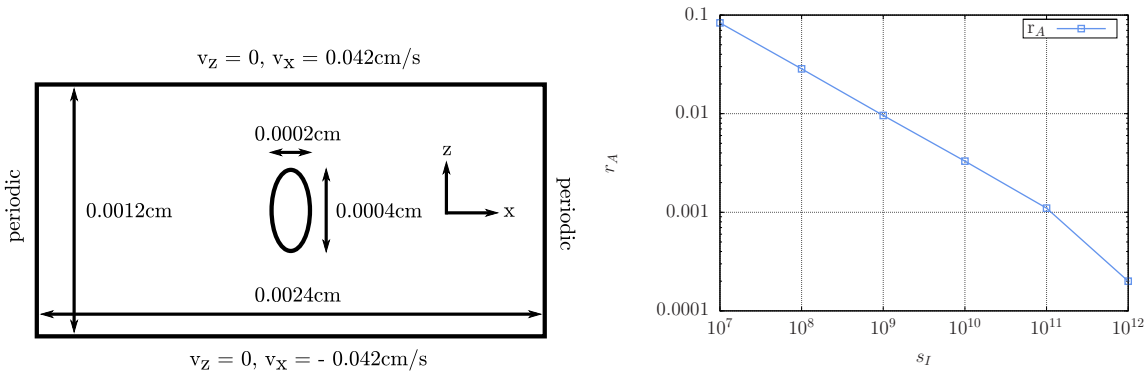
The penalty term considered in Eq. (20) becomes ineffective for the problems considered in this Section, at least for the values of the penalty parameters κ^s that enable us to converge the problem. Therefore, we consider an alternative strategy. As in the works [67, 60, 61], we change the stabilization parameters defined in Eqs. (11) and (12) near the fluid-solid interface as follows

$$\tau_M = \left(s^h(\boldsymbol{x},t) \left(\frac{4}{\Delta t^2} + \boldsymbol{v}^h \cdot \boldsymbol{G}\boldsymbol{v}^h + C_I \nu^2 \boldsymbol{G} : \boldsymbol{G} \right) \right)^{-\frac{1}{2}} , \tag{36}$$

$$\tau_C = (\tau_M \mathrm{tr}\boldsymbol{G})^{-1} . \tag{37}$$

The scalar function $s^h(\boldsymbol{x},t)$ is a dimensionless scaling factor introduced to both locally weaken the influence of the poorly approximated pressure gradient in Eq. (9) and locally enhance the penalization of volume change due to the inverse relation between τ_M and τ_C. $s^h(\boldsymbol{x},t)$ is a function that belongs to the discrete space spanned by $\{N_A^{ED}(\boldsymbol{x})\}_{A=1}^{n_{ED}}$, i.e., $s^h(\boldsymbol{x},t) = \sum_{A=1}^{n_{ED}} s_A(t) N_A^{ED}(\boldsymbol{x})$, namely, we set the control variables associated with basis functions that intersect the interface to the same value $s_I >> 1$ and the rest of control variables are set to 1. In order to find a suitable value of s_I, we measure the relative area change (r_A) of a two-dimensional elliptic capsule in shear flow for different values of s_I. The geometrical description and boundary conditions are defined in Fig. 9 a). The prescribed boundary conditions lead to a shear rate $\dot{\gamma} = 70$ s^{-1}, which is within the range found in microcirculation [47]. The remaining physical parameters are taken as follows: fluid viscosity $\mu = 0.012$ g/(cms), solid density $\rho^s = 1.025$ g/cm^3, fluid density $\rho^f = 1.1$ g/cm^3, shear modulus of the membrane $\mu_m^s = 7500$ dyn/cm^2, membrane thickness $h = 0.0000005$ cm, and $\boldsymbol{g} = (0,0)$. For the selected parameters, the capsule is supposed to undergo a tank-treading motion, i.e., the capsule membrane is supposed to adopt a nearly constant orientation with the flow direction and rotate around the interior fluid.

Regarding the spatial discretization, an Eulerian mesh with 256×128 cubic NURBS elements is used as background mesh. NURBS shape functions of second order or higher, unlike Lagrange polynomials, are



(a) Problem setup

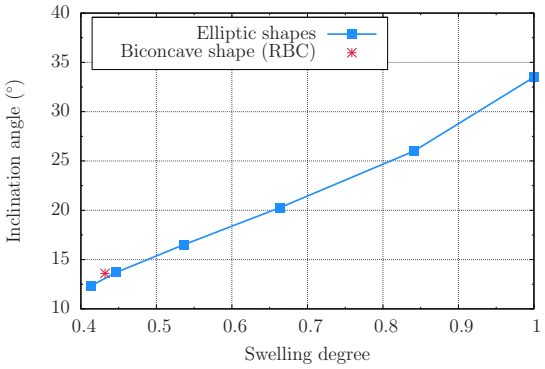
(b) Relative area change

Figure 9: *Parameter sweep of s_I so as to make sure that the incompressibility constraint is adequately enforced close to the fluid-solid interface. a) Geometrical description of the considered problem along with the boundary conditions applied. b) Relative area change of an elliptic capsule under shear flow at $t = 1.0$ s with $\Delta t = 0.0005$ s.*

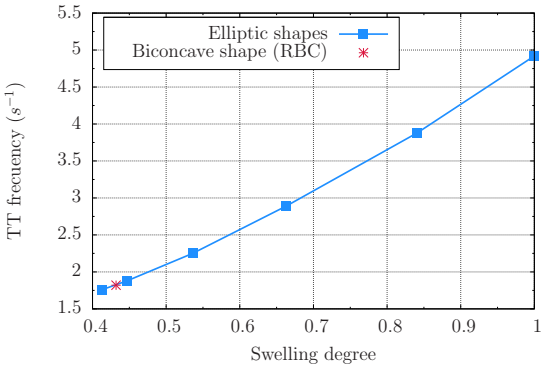
able to represent a hollow ellipse exactly, namely, we use a Lagrangian mesh with 1×208 quadratic NURBS elements to discretize the capsule. Regarding the time discretization, the time step used in the simulation is $\Delta t = 0.0005$ s. For these simulations an all the others included in this Section and in Appendix D, we have developed a MPI implementation of our NURBS-based immersed approach on top of PetIGA [36].

For high values of s_I, the capsule acquires a nearly constant inclination angle at time $t = 0.15$ s and the membrane particles have already given more than three complete turns in its TT motion at time $t = 1$ s. Fig. 9 b) shows the value of r_A for different values of s_I at time $t = 1$ s. As expected, the spurious area change decreases as we rise the dimensionless parameter s_I. Consistently with [67], we find that high errors in the incompressibility constraint may even alter the qualitative character of numerical solutions, namely, for $s_I \leq 10^5$, the capsule no longer undergoes a classical TT motion. Therefore, as a first step, it is highly recommended to check how well the incompressibility constraint is being satisfied whenever an immersed FSI approach is used.

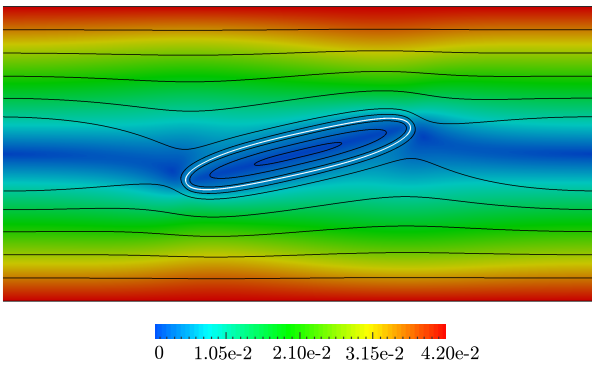
After a thorough mesh independence study detailed in Appendix D, we consider capsules with different swelling degrees (Δ_m) and embed them in the shear flow defined in Fig. 9 a). In our two-dimensional setting, Δ_m is defined as the ratio of the area enclosed by the membrane (A_e) to the area of a circle with the membrane perimeter (P_c), that is, $\Delta_m = 4\pi A_e / P_c^2$. In order to create capsules with different swelling degrees, the long axis of the capsule is modified and the remaining geometrical and mechanical parameters of the capsule are maintained the same; namely, we consider the next values for the long axis of the capsules: 0.0002 cm, 0.0004 cm, 0.0006 cm, 0.0008 cm, 0.0010 cm, and 0.0011 cm. In these simulations, s_I was set to 10^{10}. The two main physical quantities that characterize a TT motion are the inclination angle of the membrane with the flow direction (θ) and the average frecuency of the TT motion (ω_{TT}), which are plotted for the different



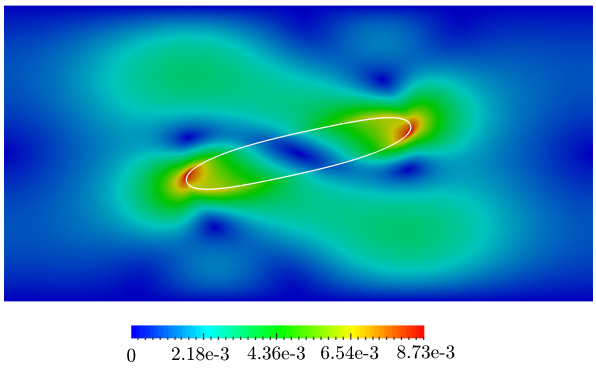
(a) θ



(b) ω_{TT}



(c) $|\boldsymbol{v}^h|$

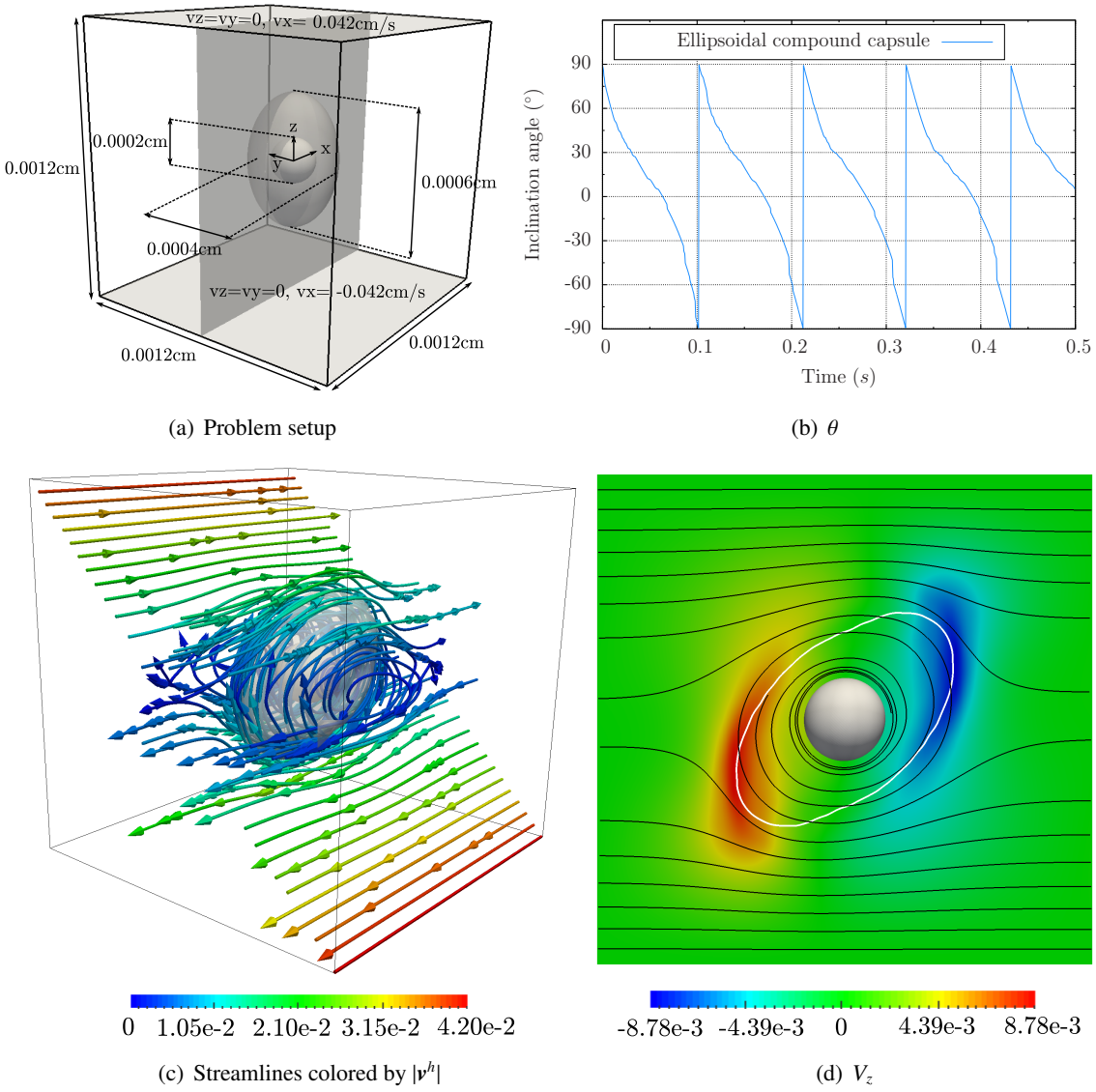


(d) $|\boldsymbol{v}^h - \boldsymbol{v}_{shear}|$

Figure 10: *Capsules with different shapes under pure shear flow. (a) Variation of the inclination angle of a capsule as a function of the swelling degree. (b) Vartiation of the tank-treading frecuency of a capsule as a function of the swelling degree. (c) Velocity magnitude along with the flow streamlines at time $t = 0.5$ s for the capsule with initial biconcave shape. The white line represents the capsule and the black lines denote the streamlines. There is a vortex whose center is located at the center of gravity of the capsule. (d) Perturbations introduced by the biconcave capsule with respect to the shear component at time $t = 0.5$ s.*

capsules in Figs. 10 a) and b), respectively. In agreement with [68], both θ and ω_{TT} increase with the value of the swelling degree. Additionally, the biconcave shape that RBCs exhibit in a quiescent fluid [115] is considered as initial shape for a capsule. The obtained θ and ω_{TT} for the biconcave shape are also plotted in Figs. 10 a) and b), respectively. Figs. 10 a) and b) suggest that the swelling degree keeps characterizing quite accurately the behavior of capsules in the nonlinear regime independently of their initial shape. Fig. 10 c) shows the velocity magnitude ($|\boldsymbol{v}^h|$) along with the streamlines for the initially biconcave-shaped capsule at time $t = 0.5$ s. Note how the streamlines near the boundary of the capsule are parallel to the membrane due to its tank-treading motion. Fig. 10 d) draws the velocity field obtained from our FSI simulation subtracting the shear component, i.e., $|\boldsymbol{v}^h - \boldsymbol{v}_{shear}|$ with the aim of showing how the capsule alters the flow conditions.

In Appendix D, the dynamics and rheology of capsules in parabolic flow are studied, finding good



(a) Problem setup (b) θ

(c) Streamlines colored by $|\boldsymbol{v}^h|$ (d) V_z

Figure 11: *A compound capsule exhibits a TU motion in a three-dimensional setting. (a) Geometrical description of the problem along with Dirichlet boundary conditions applied on the two shaded faces of the cube, periodic boundary conditions are applied in the other four faces of the cube. The shaded plane, i.e., the plane* $y = 0$ *is used to plot the z component of the velocity in Fig. 11 d). (b) Time evolution of the inclination angle. (c) Streamlines colored by the velocity magnitude at time* $t = 0.23$ *s. A transparent membrane is plotted in order to see the streamlines and the inclusion inside of it. The boundary of* Ω *is represented by black lines. (d) Velocity in z direction on the plane indicated in Fig. 11 a) at time* $t = 0.23$ *s along with the streamlines.*

agreement with RBC experiments.

In this Section and in Appendix D, we present compound capsules as a numerical proxy for cells with nucleus such as, e.g. white blood cells and circulating tumor cells. The membrane is modeled as a capsule and the nucleus is modeled as a deformable solid. The cytosol behaves as a Newtonian fluid assumed to have

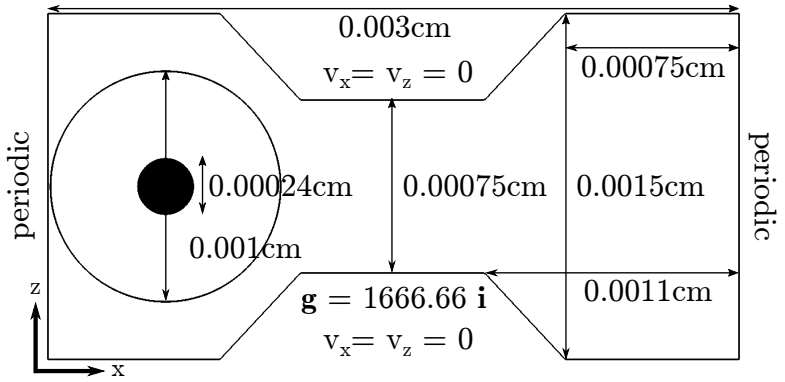


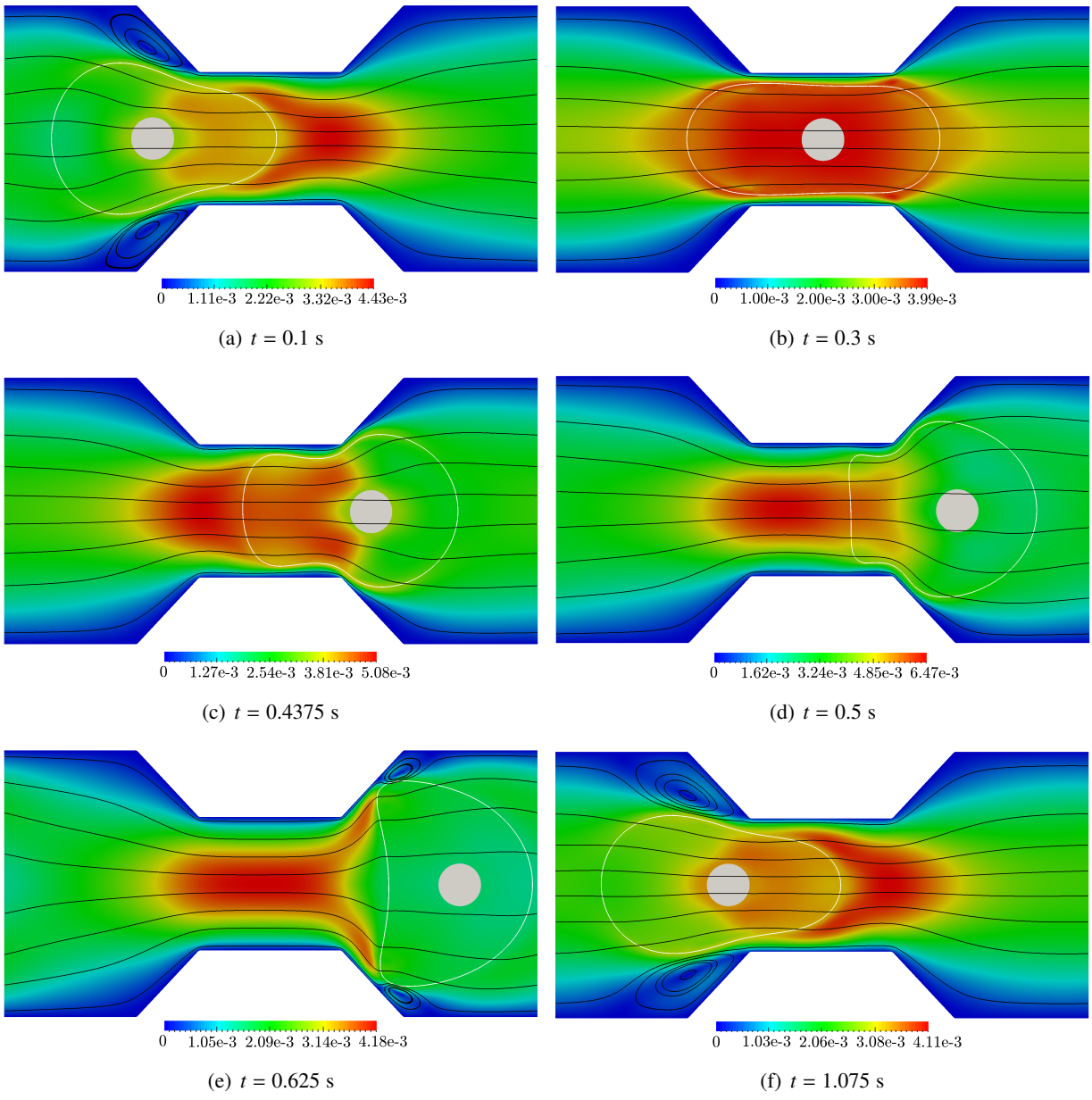
Figure 12: *(Color online) Circulating tumor cell in a channel narrowing. Problem description of our two-dimensional setting.*

the same viscosity than blood plasma for simplicity. As in [120], the organelles and fibers are neglected.

In Appendix D, the dynamics of a compound capsule in shear flow are studied in a two-dimensional setting. The behavior of our compound capsule is consistent with the results of [120], namely, as the size of the nucleus gets bigger for a certain membrane, both the inclination angle and the TT frecuency of the TT motion undergone by the compound capsule decrease.

Additionally, we consider a compound capsule in a three-dimensional setting under shear flow. The computational domain and boundary conditions are described in Fig. 11 a). The membrane is a hollow ellipsoid with semiaxes $a = 0.0003$ cm, $b = 0.0002$ cm, and $c = 0.0002$ cm. The inclusion is a sphere with a radius $r = 0.0001$ cm. The compound capsule is placed at the center of the domain with initial orientation indicated in Fig. 11 a). The remaining physical parameters are taken as follows: fluid viscosity $\mu = 0.012$ g/(cms), solid density $\rho^s = 1.025$ g/cm^3, fluid density $\rho^f = 1.1$ g/cm^3, shear modulus of the membrane $\mu_m^s = 7500$ dyn/cm^2, membrane thickness $h = 0.0000005$ cm, shear modulus of the inclusion $\mu_i^s = 40$ dyn/cm^2, and $\boldsymbol{g} = (0, 0)$. Regarding the spatial discretization, an Eulerian mesh with 262144 quadratic NURBS elements is used to represent the cube. Unlike Lagrange polynomials, NURBS functions of second order are able to represent exactly both a hollow ellipsoid and a bulky sphere, namely, we use meshes with 3072 and 7168 quadratic NURBS elements, respectively. In this computation, s_I was set to 10^9. In this simulation, both the membrane and the inclusion perform a tumbling motion, i.e., they rotate as a whole around their center of gravity. The main physical quantity that characterizes a TU motion is its average frequency (ω_{TU}). In this case, both he membrane and the inclusion have the same TU frequency, namely, $\omega_{TU} = 4.68$ s^{-1}. In Fig. 11 b), we show how the inclination angle varies in time. Fig. 11 c) displays the streamlines colored by the velocity magnitude in both a diagonal plane of the cube and close to the compound capsule at time $t = 0.23$ s. Fig. 11 d) shows the z component of the velocity field on the plane indicated in Fig. 11 a), this component is responsible for the TU motion of the membrane.

Finally, a circulating tumor cell, which is a cell with nucleus as WBCs, is forced to pass through a



(a) t = 0.1 s (b) t = 0.3 s

(c) t = 0.4375 s (d) t = 0.5 s

(e) t = 0.625 s (f) t = 1.075 s

Figure 13: *Snapshots of the velocity magnitude along with the streamlines for a CTC passing through a tube narrowing. Both the highest flow velocity and deformation of the membrane appear when the CTC is leaving the narrowing. Recirculation regions appear when the CTC is both entering and leaving the narrowing as shown in a) and e), respectively.*

channel narrowing. The CTC is modeled using a compound capsule. The membrane is a hollow circle with radius $r_m = 0.0005$ cm and thickness $h = 0.0000005$ cm. The inclusion is a circle of radius $r_i = 0.00012$ cm. The computational domain Ω is depicted in Fig. 12 along with the boundary conditions applied. The flow is driven by the pressure gradient $\boldsymbol{g} = (1666.66\ \text{dyn/cm}^3, 0\ \text{dyn/cm}^3)$. The remaining physical parameters are the following: fluid viscosity $\mu = 0.012$ g/(cms), solid density $\rho^s = 1.025\ \text{g/cm}^3$, fluid density $\rho^f = 1.1\ \text{g/cm}^3$, shear modulus of the membrane $\mu_m^s = 7500\ \text{dyn/cm}^2$, and shear modulus of the inclusion $\mu_i^s = 40\ \text{dyn/cm}^2$. A background mesh with 384×192 cubic NURBS elements is used to represent the channel. Lagrangian meshes with 1×832 and 40×192 quadratic NURBS elements are used to represent the membrane and the inclusion, respectively. The time step used in the simulation is $\Delta t = 0.00025$ s. In this simulation, s_I was set to 10^9. Fig. 13 plots the velocity magnitude along with the streamlines at different time steps, these snapshots describe how the CTC deforms in order to go through the channel narrowing. Both the highest strains of the cell and the highest velocity of the flow appear when the CTC is leaving the narrowing as can be seen in Fig. 13 d). The CTC creates two recirculation regions when it is entering the narrowing as shown in Fig. 13 a). In an analogous way, recirculation regions appear as well when the CTC is leaving the narrowing as shown in Fig. 13 e). Due to the symmetry of this FSI simulation, the membrane does not undergo TT motion in any moment since a TT motion is only possible in an asymmetric FSI simulation. In Fig. 13 f), the CTC has already given a complete turn to the computational domain and as a consequence of the periodic boundary conditions, it is about to enter in the narrowing again.

Contributions

The most significant contributions of this work, which are about to be submitted as a research paper in [30], are:

1. Capsules, modeled as solid-shell NURBS elements, are proposed as a cell mimic for red blood cells. The dynamics of these capsules are able to reproduce the main motions and shapes that red blood cells exhibit in both shear and parabolic flows provided that the incompressibility constraint near the fluid-solid interface is adequately satisfied.

2. The interaction of red blood cells in parabolic flows is studied. Rheological properties of microscale blood flow such as, e.g., Fåhræus and Fåhræus-Lindqvist effects are captured in our numerical experiments.

3. Hyperelastic compound capsules are presented as a numerical proxy for cells with nucleus. We study their dynamics in shear flow and find analogous behavior to that found for compound vesicles in [120]. Moreover, we consider the dynamics of an ellipsoidal compound capsule in a three-dimensional setting which describes a tumbling motion.

4. The ability of CTCs to go through a channel's narrowing is analyzed, this property of CTCs is attracting a remarkable attention in the experimental community since it can be used to sort CTCs from blood samples [134].

Open lines for research

The main open ideas derived from this work are:

1. Studying the interactions of red blood cells with cells with nucleus. Both white blood cells and circulating tumor cells are known to undergo margination in parabolic flows, i.e., they tend to go toward the wall of the vessel. Simulating this phenomenon in order to elucidate how margination is affected by other parameters such as, e.g., hematocrit, mean shear rate, and vessel diameter is an interesting topic.

2. Most immersed approaches applied to microcirculation neglect the inertial terms as an initial step to develop the method due to the low Reynolds numbers found in this field, this is the case of immersed approaches based on boundary integral formulations. However, recent experiments conclude that the inertial effects have an important role in microcirculation, particularly when the vessel is curved [59, 129]. We hope to address this type of problems in the near future.

3. Although we have focused on WBCs and CTCs, hyperelastic compound capsules may also be used to predict the behavior of composite core-shell microparticles, which can in turn shed light into the mechanical behaviour of analagous biomaterials, such as liposomes [76].

8 Collaborations in other works

1. Arbitrary-degree T-splines for isogeometric analysis of fully-nonlinear Kirchhoff-Love shells [33]. Note that the ability of creating analysis-suitable T-splines surfaces of arbitrary-degree, which was developed in this work, is closely linked to some of the open research lines indicated in Section 7, namely, the use of vesicles that needs a smooth surface representation of cells in order to evaluate its curvature.

2. Hybrid-degree weighted T-splines and their application in isogeometric analysis [83]. This work gives initial steps so as to obtain splines with both local h- and p-refinement.

3. Analysis suitable T-splines of arbitrary degree as a basis for variational collocation [34]. With respect to the work published in [32] where Greville points were used as collocation points, the new set of collocation points introduced in this work is able to raise the convergence rate from $p-1$ to p when p is odd.

4. A variational collocation approach for Kirchhoff-Love shells [35]. In this work, the fourth-order partial differential equations that govern Kirchhoff-Love shells are solved in strong form for the first time.

9 Conclusions

This Section summarizes the main contributions and research ideas of this thesis. We have developed mesh-based immersed approaches for fluid-structure interaction with the following characteristics:

- Both the fluid and the solid are assumed to be incompressible, which is a case of special interest in various problems of biomechanics [50]. Besides of that, we work with Newtonian viscous fluids and nonlinear hyperelastic solids. Nevertheless, other material models for the solid could be easily considered since the algorithms developed in this thesis are largely independent of the particular material used for the solid.

- The three unknows of our formulation are the Eulerian velocity, the Eulerian pressure, and the Lagrangian displacement. These three unknows are associated with the linear momentum balance equation, the mass conservation equation and the kinematic equation that relates the Lagrangian displacement with the Eulerian velocity, respectively. The first two equations are solved using a variational multiscale technique and the latter is solved using isogeometric collocation, which leads to a hybrid variational-collocation approach.

- Regarding spatial discretization, a background mesh is defined in the whole computational domain and a Lagrangian mesh is tailored to the region occupied by each solid. These meshes are initially built using NURBS, which can be used in two- and three-dimensional settings, but do not support local h-refinement. Then, the algorithms are generalized in order to use ASTS as basis functions, which support local h-refinement, but are only defined in two-dimensional settings and for surface representations up to now. Regarding the time discretization and coupling strategy, we use a block-iterative technique based on the generalized-α method, thus obtaining a fully-implicit and second-order accurate method. We validate our methodology comparing the computed terminal velocity of a free-falling bulky rigid solid with its theoretical value, excellent agreement is found if the meshes are fine enough.

- The data transfer between the background mesh and the Lagrangian meshes is performed without introducing discretized delta functions. The basis functions of the meshes are used to do so. Moreover, we proceed in such a way that no projection is needed, thus eliminating the errors associated with this procedure.

- Lagrange polynomials, which are the standard basis functions in the finite element method, have C^0 inter-element continuity for all degrees. In contrast, NURBS and ASTS are capable of getting C^{p-1} inter-element continuity, which has at least two important advantages in our immersed approaches. Firstly, the higher inter-element continuity of the Lagrangian meshes enable us to deal with larger strains in these meshes, thus enhancing the robustness of the overall algorithm. Secondly, the higher inter-element continuity of the background mesh alleviates the issue of suboptimal quadrature rules that appears in immersed FSI methods.

- We emphasize on the importance of adequately satisfying the incompressibility constraint close to the fluid-solid interface. We believe this issue deserves further attention in the community of immersed approaches for fluid-structure interaction since it may lead to a glaring error in simulations.
- Our NURBS-based code was developed on top of the PetIGA framework [36, 22, 123], which adds the NURBS discretization capabilities and integration of forms to the scientific library PETSc [8, 7]. As an initial step, a serial code was done in order to validate the formulation. Then, the code was extended in order to perform MPI computations. This last step is quite challenging due to the fact that there are several Lagrangian meshes moving freely on top of the background mesh and these meshes need to communicate with each other. PetIGA has proven to be highly efficient in [36] and is able to maintain the high continuity of the basis functions between processors. However, all the data structures used to assemble the system are developed for meshes with tensor product structure. Therefore, if the mesh does not have a tensor product structure as in the case of ASTS, PetIGA does not offer any support to the user. As a consequence, for the ASTS simulations presented in this thesis, a serial code was developed directly on top of PETSc without using PetIGA.

The spline-based approaches proposed in this thesis can be applied to a large variety of interesting FSI problems in biomechanics and other fields. Here, we have focused on cell-scale blood flow. The main simulation results are the following:

- Hyperelastic capsules, modeled as solid-shell NURBS elements, are used as our cell mimic for red blood cells. As long as the incompressibility constraint is properly enforced near the fluid-solid interface, the dynamics of capsules are able to reproduce the main motions and shapes observed in experiments with red blood cells in both shear and parabolic flows. Moreover, the qualitative behavior of various rheological properties of red blood cells in parabolic flow is captured in our simulations.
- Models for cells with nucleus such as, e.g., white blood cells and circulating tumor cells, are currently much less advanced than those for red blood cells. In this thesis, we present hyperelastic compound capsules for modeling this type of cells. The dynamics of compound capsules in shear flow are studied in a two-dimensional setting. In order to analyze the influence of the nucleus, we consider compound capsules with different inclusion size and same membrane. Consistently with the results for compound vesicles [120], as the size of the inclusion increases, the inclination angle and TT frecuency of the TT motion undergone by the compound capsules decrease. Additionally, taking advantage of the geometrical flexibility of our immersed approaches, we study how a circulating tumor cell manages to pass through channel narrowings. The simulation reveals that the highest deformation of the CTC and the highest velocity of the flow appear when the CTC is about to leave the narrowing.

The work accomplished in this thesis leaves various open research lines for the future. We proceed to summarize the following lines:

- There are various appealing phenomena such as, e.g., margination of cells with nucleus in parabolic flows and sorting of cells with nucleus from blood samples, which involve the interaction of red blood cells with either white blood cells or circulating tumor cells. In some of these phenomena, inertial effects play a main role [59, 129]. Our immersed approaches take into account inertial effects which need to be neglected in other types of immersed approaches as those based on boundary integral methods [49]. Therefore, our formulation seems very suitable to tackle these applications and we hope to do so in the near future.

- Different models could be considered to represent the membrane of cells. For example, vesicles, which are based on a surface representation, need to compute the curvature, i.e., second spatial derivatives are needed in the formulation. The use of both NURBS and ASTS surface representations would be very appropriate here due to their capability to construct spaces with global C^1 or higher continuity. Moreover, it seems that a precise comparison between vesicles and capsules to establish the main differences in its behavior is lacking in the literature.

- Divergence-conforming B-splines [44] can be considered for solving the linear momentum balance equation and the mass conservation equation instead of using the VMS technique. In general, this alternative is appealing for applications with low Reynolds numbers, which is the case in cell-scale blood flow. Additionally, this formulation is able to obtain pointwise mass conservation as long as the residual is properly converged which could fix the issue of poor pressure approximation near the fluid-solid interface. However, the convergence of the residual, which is always challenging to achieve in a saddle point problem, will be even more difficult to obtain as solids are embedded in the domain.

- Developing adaptive h-refinement capabilities for the background mesh in order to have enhanced resolution near the fluid-solid interface is a promising research direction as well. As long as the adaptive part is coded efficiently, this could speed up simulations in a three-dimensional setting.

References

[1] D. Abreu, M. Levant, V. Steinberg, and U. Seifert. Fluid vesicles in flow. *Advances in colloid and interface science*, 208:129–141, 2014.

[2] I. Akkerman, Y. Bazilevs, V. M. Calo, T. J. R. Hughes, and S. Hulshoff. The role of continuity in residual-based variational multiscale modeling of turbulence. *Computational Mechanics*, 41:371–378, 2008.

[3] F. Auricchio, L. Beirao da Veiga, T. Hughes, A. Reali, and G. Sangalli. Isogeometric collocation for elastostatics and explicit dynamics. *Computer Methods in Applied Mechanics and Engineering*, 249:2–14, 2012.

[4] F. Auricchio, L. Beirao Da Veiga, T. J. R. Hughes, A. Reali, and G. Sangalli. Isogeometric collocation methods. *Mathematical Models and Methods in Applied Sciences*, 20:2075–2107, 2010.

[5] I. Babuška. Error-bounds for finite element method. *Numerische Mathematik*, 16(4):322–333.

[6] P. Bagchi and R. M. Kalluri. Dynamics of nonspherical capsules in shear flow. *Physical Review E*, 80(1):016307, 2009.

[7] S. Balay, M. F. Adams, J. Brown, P. Brune, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, and H. Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 3.4, Argonne National Laboratory, 2013.

[8] S. Balay, M. F. Adams, J. Brown, P. Brune, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, and H. Zhang. PETSc Web page. `http://www.mcs.anl.gov/petsc`, 2014.

[9] H. Basu, A. K. Dharmadhikari, J. A. Dharmadhikari, S. Sharma, and D. Mathur. Tank treading of optically trapped red blood cells in shear flow. *Biophysical journal*, 101(7):1604–1612, 2011.

[10] Y. Bazilevs, V. Calo, J. Cottrell, J. Evans, T. J. R. Hughes, S. Lipton, M. Scott, and T. Sederberg. Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering*, 199:229–263, 2010.

[11] Y. Bazilevs, V. M. Calo, J. A. Cottrell., T. J. R. Hughes, A. Reali, and G. Scovazzi. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 197:173–201, 2007.

[12] Y. Bazilevs, M.-C. Hsu, I. Akkerman, S. Wright, K. Takizawa, B. Henicke, T. Spielman, and T. Tezduyar. 3D simulation of wind turbine rotors at full scale. part I: Geometry modeling and aerodynamics. *International Journal for Numerical Methods in Fluids*, 65:207–235, 2011.

[13] Y. Bazilevs, M.-C. Hsu, J. Kiendl, R. Wuchner, and K.-U. Bletzinger. 3D simulation of wind turbine rotors at full scale. part ii: Fluid-structure interaction modeling with composite blades. *International Journal for Numerical Methods in Fluids*, 65:236–253, 2011.

[14] Y. Bazilevs, M.-C. Hsu, and M. Scott. Isogeometric fluid-structure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines. *Computer Methods in Applied Mechanics and Engineering*, 249-252:28–41, 2012.

[15] Y. Bazilevs, M.-C. Hsu, Y. Zhang, W. Wang, T. Kvamsdal, S. Hentschel, and J. Isaksen. Computational vascular fluid-structure interaction: Methodology and application to cerebral aneurysms. *Biomechanics and Modeling in Mechanobiology*, 9:481–498, 2010.

[16] Y. Bazilevs, M.-C. Hsu, Y. Zhang, W. Wang, X. Liang, T. Kvamsdal, R. Brekken, and J. Isaksen. A fully-coupled fluid-structure interaction simulation of cerebral aneurysms. *Computational Mechanics*, 46(1):3–16, 2010.

[17] Y. Bazilevs, K. Takizawa, and T. E. Tezduyar. *Computational fluid-structure interaction: methods and applications*. John Wiley & Sons, 2012.

[18] L. Beirao da Veiga, A. Buffa, D. Cho, and G. Sangalli. Analysis-suitable t-splines are dual-compatible. *Computer methods in applied mechanics and engineering*, 249:42–51, 2012.

[19] L. Beirao da Veiga, A. Buffa, G. Sangalli, and R. Vazquez. Analysis suitable T-splines of arbitrary degree: Definition, linear independence, and approximation properties. *Mathematical Models and Methods in Applied Sciences*, 23(11):1979–2003, 2013.

[20] D. Benson, Y. Bazilevs, M.-C. Hsu, and T. J. R. Hughes. A large deformation, rotation-free, isogeometric shell. *Computer Methods in Applied Mechanics and Engineering*, 200(13):1367–1378, 2011.

[21] D. Benson, S. Hartmann, Y. Bazilevs, M.-C. Hsu, and T. J. R. Hughes. Blended isogeometric shells. *Computer Methods in Applied Mechanics and Engineering*, 255:133 – 146, 2013.

[22] M. Bernal, V. M. Calo, N. Collier, G. Espinosa, F. Fuentes, and J. Mahecha. Isogeometric analysis of hyperelastic materials using petiga. *Procedia Computer Science*, 18:1604–1613, 2013.

[23] R. Beyer Jr. A computational model of the cochlea using the immersed boundary method. *Journal of Computational Physics*, 98:145–162, 1992.

[24] M. Borden, C. Verhoosel, M. Scott, T. Hughes, and C. Landis. A phase-field description of dynamic brittle fracture. *Computer Methods in Applied Mechanics and Engineering*, 217-220:77–95, 2012.

[25] A. Bressan, A. Buffa, and G. Sangalli. Characterization of analysis-suitable T-splines. *Computer Aided Geometric Design*, 39:17–49, 2015.

[26] F. Brezzi. On the existence, uniqueness and approximation of saddle-point problems arising from lagrangian multipliers. *ESAIM: Mathematical Modelling and Numerical Analysis*, 8:129–151, 1974.

[27] J. Bueno, C. Bona-Casas, Y. Bazilevs, and H. Gomez. Interaction of complex fluids and solids: theory, algorithms and application to phase-change-driven implosion. *Computational Mechanics*, 55:1105–1118, 2015.

[28] A. Buffa, D. Cho, and G. Sangalli. Linear independence of the T-spline blending functions associated with some particular T-meshes. *Computer Methods in Applied Mechanics and Engineering*, 199(23):1437–1445, 2010.

[29] H. Casquero, C. Bona-Casas, and H. Gomez. A NURBS-based immersed methodology for fluid-structure interaction. *Computer Methods in Applied Mechanics and Engineering*, 284:943–970, 2015.

[30] H. Casquero, C. Bona-Casas, and H. Gomez. NURBS-based numerical proxies for red blood cells and circulating tumor cells in microscale blood flow. *Preprint submitted to Computer Methods in Applied Mechanics and Engineering*, 2016.

[31] H. Casquero, L. Liu, C. Bona-Casas, Y. Zhang, and H. Gomez. A hybrid variational-collocation immersed method for fluid-structure interaction using unstructured T-splines. *International Journal for Numerical Methods in Engineering*, 105(11):855–880, 2016.

[32] H. Casquero, L. Liu, Y. Zhang, A. Reali, and H. Gomez. Isogeometric collocation using analysis-suitable T-splines of arbitrary degree. *Computer Methods in Applied Mechanics and Engineering*, 301:164 – 186, 2016.

[33] H. Casquero, L. Liu, Y. Zhang, A. Reali, J. Kiendl, and H. Gomez. Arbitrary-degree T-splines for isogeometric analysis of fully nonlinear Kirchhoff-Love shells. *Under review*, 2016.

[34] H. Casquero, X. Wei, Y. Zhang, L. De Lorenzis, and H. Gomez. Analysis suitable T-splines of arbitrary degree as a basis for variational collocation. *In preparation*, 2016.

[35] H. Casquero, X. Wei, Y. Zhang, A. Reali, J. Kiendl, and H. Gomez. A variational collocation approach for Kirchhoff-Love shells. *In preparation*, 2016.

[36] N. Collier, L. Dalcin, and V. M. Calo. Petiga: High-performance isogeometric analysis. *arXiv preprint:1305.4452*, 2013.

[37] J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs. *Isogeometric Analysis Toward Integration of CAD and FEA*. Wiley, 2009.

[38] J. A. Cottrell, T. J. R. Hughes, and A. Reali. Studies of refinement and continuity in isogeometric structural analysis. *Computer Methods in Applied Mechanics and Engineering*, 196(41-44):4160–4183, 2007. cited By (since 1996)186.

[39] G. Danker, P. M. Vlahovska, and C. Misbah. Vesicles in poiseuille flow. *Physical review letters*, 102(14):148102, 2009.

[40] L. De Lorenzis, J. Evans, T. Hughes, and A. Reali. Isogeometric collocation: Neumann boundary conditions and contact. *Computer Methods in Applied Mechanics and Engineering*, 284:21–54, 2015.

[41] R. Dhote, H. Gomez, R. Melnik, and J. Zu. 3D coupled thermo-mechanical phase-field modeling of shape memory alloy dynamics via isogeometric analysis. *Computers & Structures*, 2015.

[42] R. Dillon, L. Fauci, A. Fogelson, and D. Gaver III. Modeling biofilm processes using the immersed boundary method. *Journal of Computational Physics*, 129:57–73, 1996.

[43] R. Dimitri, L. D. Lorenzis, M. Scott, P. Wriggers, R. Taylor, and G. Zavarise. Isogeometric large deformation frictionless contact using T-splines. *Computer Methods in Applied Mechanics and Engineering*, 269:394–414, 2014.

[44] J. A. Evans and T. J. Hughes. Isogeometric divergence-conforming B-splines for the unsteady Navier–Stokes equations. *Journal of Computational Physics*, 241:141–167, 2013.

[45] C. Farhat, K. G. van der Zee, and P. Geuzaine. Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity. *Computer methods in applied mechanics and engineering*, 195:1973–2001, 2006.

[46] L. Fauci and C. Peskin. A computational model of aquatic animal locomotion. *Journal of Computational Physics*, 77:85–108, 1988.

[47] D. A. Fedosov, M. Peltomäki, and G. Gompper. Deformation and dynamics of red blood cells in flow through cylindrical microchannels. *Soft Matter*, 10(24):4258–4267, 2014.

[48] T. Fischer, M. Stohr-Lissen, and H. Schmid-Schonbein. The red cell as a fluid droplet: tank tread-like motion of the human erythrocyte membrane in shear flow. *Science*, 202:894–896, 1978.

[49] J. B. Freund. Numerical simulation of flowing blood cells. *Annual review of fluid mechanics*, 46:67–95, 2014.

[50] Y. C. Fung. *Biomechanics*. Springer, New York, 1990.

[51] M. Gay, L. Zhang, and W. K. Liu. Stent deployment using immersed finite element method. *Comput. Methods Appl. Mech. Engrg.*, 2005. cited By (since 1996)1.

[52] R. Glowinski, T. Pan, T. Hesla, D. Joseph, and J. Periaux. A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: application to particulate flow. *Journal of Computational Physics*, 169(2):363–426, 2001.

[53] H. Goldsmith and J. Marlow. Flow behaviour of erythrocytes. i. rotation and deformation in dilute suspensions. *Proceedings of the Royal Society of London B: Biological Sciences*, 182(1068):351–384, 1972.

[54] H. Gomez, V. M. Calo, Y. Bazilevs, and T. J. R. Hughes. Isogeometric analysis of the Cahn-Hilliard phase-field model. *Computer Methods in Applied Mechanics and Engineering*, 197(49-50):4333–4352, 2008.

[55] H. Gomez and L. De Lorenzis. The variational collocation method. *submitted for publication*, 2016.

[56] H. Gomez, A. Reali, and G. Sangalli. Accurate, efficient, and (iso)geometrically flexible collocation methods for phase-field models. *Journal of Computational Physics*, 262:153–171, 2014.

[57] J. Hellums. 1993 whitaker lecture: Biorheology in thrombosis research. *Annals of Biomedical Engineering*, 22(5):445–455, 1994.

[58] C. Hesch, A. Gil, A. Arranz Carreo, and J. Bonet. On continuum immersed strategies for fluid-structure interaction. *Computer Methods in Applied Mechanics and Engineering*, 247-248:51–64, 2012.

[59] H. W. Hou, M. E. Warkiani, B. L. Khoo, Z. R. Li, R. A. Soo, D. S.-W. Tan, W.-T. Lim, J. Han, A. A. S. Bhagat, and C. T. Lim. Isolation and retrieval of circulating tumor cells using centrifugal forces. *Scientific Reports*, 3:1259, 2013.

[60] M.-C. Hsu, D. Kamensky, Y. Bazilevs, M. Sacks, and T. J. R. Hughes. Fluid-structure interaction analysis of bioprosthetic heart valves: Significance of arterial wall deformation. *Computational Mechanics*, 54:1055–1071, 2014.

[61] M.-C. Hsu, D. Kamensky, F. Xu, J. Kiendl, C. Wang, M. Wu, J. Mineroff, A. Reali, Y. Bazilevs, and M. Sacks. Dynamic and fluid-structure interaction simulations of bioprosthetic heart valves using parametric design with T-splines and Fung-type material models. *Computational Mechanics*, 55(6):1211–1225, 2015.

[62] T. J. R. Hughes. *The finite element method: linear static and dynamic finite element analysis*. Courier Corporation, 2012.

[63] T. J. R. Hughes, J. A. Cottrell, and Y. Bazilevs. Isogeometric analysis CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computacional Methods in Applied Mechanics and Engineering*, 194:4135–4195, 2005.

[64] T. J. R. Hughes, G. Feijóo, L. Mazzei, and J.-B. Quincy. The variational multiscale method - A paradigm for computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, 166:3–24, 1998.

[65] T. J. R. Hughes and G. Sangalli. Variational multiscale analysis: The fine-scale green's function, projection, optimization, localization, and stabilized methods. *SIAM Journal on Numerical Analysis*, 45:539–557, 2007.

[66] K. Jansen, C. Whiting, and G. Hulbert. Generalized-α method for integrating the filtered navier-stokes equations with a stabilized finite element method. *Computer Methods in Applied Mechanics and Engineering*, 190:305–319, 2000.

[67] D. Kamensky, M.-C. Hsu, D. Schillinger, J. A. Evans, A. Aggarwal, Y. Bazilevs, M. S. Sacks, and T. J. R. Hughes. An immersogeometric variational framework for fluid-structure interaction: Application to bioprosthetic heart valves. *Computer Methods in Applied Mechanics and Engineering*, 284:1005–1053, 2015.

[68] B. Kaoui, J. Harting, and C. Misbah. Two-dimensional vesicle dynamics under shear flow: Effect of confinement. *Physical Review E*, 83(6):066319, 2011.

[69] S. R. Keller and R. Skalak. Motion of a tank-treading ellipsoidal particle in a shear flow. *Journal of Fluid Mechanics*, 120:27–47, 1982.

[70] G. Késmárky, P. Kenyeres, M. Rábai, and K. Tóth. Plasma viscosity: a forgotten variable. *Clinical hemorheology and microcirculation*, 39(1–4):243–246, 2008.

[71] J. Kiendl, F. Auricchio, L. Beirão da Veiga, C. Lovadina, and A. Reali. Isogeometric collocation methods for the Reissner-Mindlin plate problem. *Computer Methods in Applied Mechanics and Engineering*, 284:489 – 507, 2015.

[72] J. Kiendl, F. Auricchio, T. J. R. Hughes, and A. Reali. Single-variable formulations and isogeometric discretizations for shear deformable beams. *Computer Methods in Applied Mechanics and Engineering*, 284:988 – 1004, 2015.

[73] J. Kiendl, Y. Bazilevs, M.-C. Hsu, R. Wüchner, and K.-U. Bletzinger. The bending strip method for isogeometric analysis of Kirchhoff–Love shell structures comprised of multiple patches. *Computer Methods in Applied Mechanics and Engineering*, 199:2403–2416, 2010.

[74] J. Kiendl, K.-U. Bletzinger, J. Linhard, and R. Wuchner. Isogeometric shell analysis with Kirchhoff-Love elements. *Computer Methods in Applied Mechanics and Engineering*, 198:3902–3914, 2009.

[75] J. Kiendl, M.-C. Hsu, M. C. Wu, and A. Reali. Isogeometric Kirchhoff–Love shell formulations for general hyperelastic materials. *Computer Methods in Applied Mechanics and Engineering*, 291:280–303, 2015.

[76] T. Kong, L. Wang, H. M. Wyss, and H. C. Shum. Capillary micromechanics for core-shell particles. *Soft Matter*, 10:3271–3276, 2014.

[77] P. Koumoutsakos, I. Pivkin, and F. Milde. The fluid mechanics of cancer and its therapy. *Annual Review of Fluid Mechanics*, 45:325–355, 2013.

[78] A. Krogh. *The anatomy and physiology of capillaries*, volume 18. Yale University Press, 1922.

[79] X. Li and M. A. Scott. Analysis-suitable T-splines: characterization, refineability, and approximation. *Mathematical Models and Methods in Applied Sciences*, 24(06):1141–1164, 2014.

[80] X. Li, J. Zheng, T. W. Sederberg, T. J. R. Hughes, and M. A. Scott. On linear independence of T-spline blending functions. *Computer Aided Geometric Design*, 29:63–76, 2012.

[81] S. Lipton, J. Evans, Y. Bazilevs, T. Elguedj, and T. J. R. Hughes. Robustness of isogeometric structural discretizations under severe mesh distortion. *Computer Methods in Applied Mechanics and Engineering*, 199:357–373, 2010.

[82] J. Liu, C. M. Landis, H. Gomez, and T. J. R. Hughes. Liquid-vapor phase transition: Thermomechanical theory, entropy stable numerical formulation, and boiling simulations. *Computer Methods in Applied Mechanics and Engineering*, 2015.

[83] L. Liu, H. Casquero, H. Gomez, and Y. J. Zhang. Hybrid-degree weighted T-splines and their application in isogeometric analysis. *Computers & Fluids*, 2016.

[84] L. Liu, Y. Zhang, T. J. R. Hughes, M. A. Scott, and T. W. Sederberg. Volumetric T-spline construction using Boolean operations. *Engineering with Computers*, 30:425–439, 2014.

[85] L. Liu, Y. Zhang, Y. Liu, and W. Wang. Feature-preserving T-mesh construction using skeleton-based polycubes. *Computer-Aided Design*, 58:162–172, 2015.

[86] P. L.-F. Liu, P. Lin, K.-A. Chang, and T. Sakakiyama. Numerical modeling of wave interaction with porous structures. *Journal of waterway, port, coastal, and ocean engineering*, 125(6):322–330, 1999.

[87] W. K. Liu, S. Jun, S. Li, J. Adee, and T. Belytschko. Reproducing kernel particle methods for structural dynamics. *International Journal for Numerical Methods in Engineering*, 38:1655–1679, 1995. cited By (since 1996)604.

[88] W. K. Liu, S. Jun, and Y. F. Zhang. Reproducing kernel particle methods. *International Journal for Numerical Methods in Fluids*, 20(8-9):1081–1106, 1995. cited By (since 1996)1295.

[89] W. K. Liu, Y. Liu, D. Farrell, L. Zhang, X. Wang, Y. Fukui, N. Patankar, Y. Zhang, C. Bajaj, J. Lee, J. Hong, X. Chen, and H. Hsu. Immersed finite element method and its applications to biological systems. *Computer Methods in Applied Mechanics and Engineering*, 195:1722–1749, 2006.

[90] Y. Liu and W. K. Liu. Rheology of red blood cell aggregation by computer simulation. *Journal of Computational Physics*, 220:139–154, 2006.

[91] Y. Liu, L. Zhang, X. Wang, and W. K. Liu. Coupling of Navier-Stokes equations with protein molecular dynamics and its application to hemodynamics. *International Journal for Numerical Methods in Fluids*, 46:1237–1252, 2004.

[92] M.-A. Mader, H. Ez-Zahraouy, C. Misbah, and T. Podgorski. On coupling between the orientation and the shape of a vesicle under a shear flow. *The European Physical Journal E: Soft Matter and Biological Physics*, 22(4):275–280, 2007.

[93] D. McQueen and C. Peskin. Computer-assisted design of butterfly bileaflet valves for the mitral position. *Scandinavian Journal of Thoracic and Cardiovascular Surgery*, 19:139–148, 1985.

[94] S. Meσ^s linger, B. Schmidt, H. Noguchi, and G. Gompper. Dynamical regimes and hydrodynamic lift of viscous vesicles under shear. *Physical Review E*, 80(1):011901, 2009.

[95] C. Peskin. Flow patterns around heart valves: A numerical method. *Journal of Computational Physics*, 10:252–271, 1972.

[96] C. Peskin. Numerical analysis of blood flow in the heart. *Journal of Computational Physics*, 25(3):220–252, 1977.

[97] L. Piegl and W. Tiller. *The NURBS book*. Springer Science & Business Media, 2012.

[98] S. Piperno, C. Farhat, and B. Larrouturou. Partitioned procedures for the transient solution of coupled aroelastic problems part i: Model problem, theory and two-dimensional application. *Computer Methods in Applied Mechanics and Engineering*, 124(1-2):79–112, 1995. cited By (since 1996)190.

[99] I. Pivkin and G. Karniadakis. Accurate coarse-grained modeling of red blood cells. *Physical Review Letters*, 101(11), 2008.

[100] A. Pommella, S. Caserta, and S. Guido. Dynamic flow behaviour of surfactant vesicles under shear flow: role of a multilamellar microstructure. *Soft Matter*, 9(31):7545–7552, 2013.

[101] C. Pozrikidis. *Modeling and simulation of capsules and biological cells*. CRC Press, 2003.

[102] C. Pozrikidis. Axisymmetric motion of red blood cells through capillaries. *Phys. Fluids*, 2004.

[103] S. Ramanujan and C. Pozrikidis. Deformation of liquid capsules enclosed by elastic membranes in simple shear flow: large deformations and the effect of fluid viscosities. *Journal of Fluid Mechanics*, 361:117–143, 1998.

[104] A. Reali and H. Gomez. An isogeometric collocation approach for Bernoulli-Euler beams and Kirchhoff plates. *Computer Methods in Applied Mechanics and Engineering*, 284:623–636, 2015.

[105] F. Rioual, T. Biben, and C. Misbah. Analytical analysis of a vesicle tumbling under a shear flow. *Physical Review E*, 69(6):061914, 2004.

[106] D. F. Rogers. An introduction to NURBS: With historical perspective. Morgan Kaufmann, San Francisco, 2001.

[107] D. Schillinger, L. Ded, M. A. Scott, J. A. Evans, M. J. Borden, E. Rank, and T. J. Hughes. An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces. *Computer Methods in Applied Mechanics and Engineering*, 249252:116 – 150, 2012. Higher Order Finite Element and Isogeometric Methods.

[108] D. Schillinger, J. Evans, A. Reali, M. Scott, and T. J. R. Hughes. Isogeometric collocation: Cost comparison with Galerkin methods and extension to adaptive hierarchical NURBS discretizations. *Computer Methods in Applied Mechanics and Engineering*, 267:170–232, 2013.

[109] H. Schmid-Schönbein and R. Wells. Fluid drop-like transition of erythrocytes under shear. *Science*, 165(3890):288–291, 1969.

[110] M. Scott, X. Li, T. Sederberg, and T. J. R. Hughes. Local refinement of analysis-suitable T-splines. *Computer Methods in Applied Mechanics and Engineering*, 213-216:206–222, 2012.

[111] T. W. Sederberg, D. L. Cardon, G. T. Finnigan, N. S. North, J. Zheng, and T. Lyche. T-spline simplification and local refinement. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 276–283. ACM, 2004.

[112] T. W. Sederberg, G. T. Finnigan, X. Li, H. Lin, and H. Ipson. Watertight trimmed NURBS. In *ACM SIGGRAPH 2008 Papers*, SIGGRAPH '08, pages 79:1–79:8, New York, NY, USA, 2008. ACM.

[113] T. W. Sederberg, J. Zheng, A. Bakenov, and A. Nasri. T-splines and T-NURCCs. *ACM Trans. Graph.*, 22:477–484, 2003.

[114] R. Simpson, M. Scott, M. Taus, D. Thomas, and H. Lian. Acoustic isogeometric boundary element analysis. *Computer Methods in Applied Mechanics and Engineering*, 269:265–290, 2014.

[115] R. Skalak, A. Tozeren, R. Zarda, and S. Chien. Strain energy function of red blood cell membranes. *Biophysical Journal*, 13(3):245 – 264, 1973.

[116] K. Takizawa and T. E. Tezduyar. Computational methods for parachute fluid-structure interactions. *Archives of Computational Methods in Engineering*, 19:125–169, 2012.

[117] T. E. Tezduyar and Y. Osawa. Fluid-structure interactions of a parachute crossing the far wake of an aircraft. *Computer Methods in Applied Mechanics and Engineering*, 191:717–726, 2001.

[118] A. Tijsseling. Fluid-structure interaction in liquid-filled pipe systems: a review. *Journal of Fluids and Structures*, 10(2):109–146, 1996.

[119] R. Tran-Son-Tay and G. B. Nash. *Hemorheology and hemodynamics*. IOS Press, Amsterdam, 2007.

[120] S. K. Veerapaneni, Y.-N. Young, P. M. Vlahovska, and J. Bławzdziewicz. Dynamics of a compound vesicle in shear flow. *Physical review letters*, 106(15):158103, 2011.

[121] C. V. Verhoosel, M. A. Scott, R. de Borst, and T. J. R. Hughes. An isogeometric approach to cohesive zone modeling. *International Journal for Numerical Methods in Engineering*, 87:336–360, 2011.

[122] C. V. Verhoosel, M. A. Scott, T. J. R. Hughes, and R. de Borst. An isogeometric analysis approach to gradient damage models. *International Journal for Numerical Methods in Engineering*, 86:115–134, 2011.

[123] P. A. Vignal, N. Collier, and V. M. Calo. Phase field modeling using petiga. *Procedia Computer Science*, 18:1614–1623, 2013.

[124] G. Vilanova, I. Colominas, and H. Gomez. Capillary networks in tumor angiogenesis: From discrete endothelial cells to phase-field averaged descriptions via isogeometric analysis. *International Journal for Numerical Methods in Biomedical Engineering*, 29:1015–1037, 2013.

[125] V. Vitkova, M. Mader, and T. Podgorski. Deformation of vesicles flowing through capillaries. *EPL (Europhysics Letters)*, 68(3):398, 2004.

[126] A.-V. Vuong, C. Giannelli, B. Jüttler, and B. Simeon. A hierarchical approach to adaptive local refinement in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 200(49):3554–3567, 2011.

[127] C. Wang, M. C. Wu, F. Xu, M.-C. Hsu, and Y. Bazilevs. Modeling of a hydraulic arresting gear using fluid–structure interaction and isogeometric analysis. *Computers & Fluids*, 2015.

[128] W. Wang, Y. Zhang, L. Liu, and T. J. R. Hughes. Trivariate solid T-spline construction from boundary triangulations with arbitrary genus topology. *Computer-Aided Design*, 45(2):351–360, 2013.

[129] M. E. Warkiani, B. L. Khoo, L. Wu, A. K. P. Tay, A. A. S. Bhagat, J. Han, and C. T. Lim. Ultra-fast, label-free isolation of circulating tumor cells from blood using spiral microfluidics. *Nat. Protocols*, 11(1):134–148, 2016.

[130] J. Yan, A. Korobenko, X. Deng, and Y. Bazilevs. Computational free-surface fluid–structure interaction with application to floating offshore wind turbines. *Computers & Fluids*, 2016.

[131] L. Zhang, A. Gerstenberger, X. Wang, and W. K. Liu. Immersed finite element method. *Computer Methods in Applied Mechanics and Engineering*, 193:2051–2067, 2004.

[132] Y. Zhang, W. Wang, and T. J. R. Hughes. Solid T-spline construction from boundary representations for genus-zero geometry. *Computer Methods in Applied Mechanics and Engineering*, 249-252:185–197, 2012.

[133] Y. Zhang, W. Wang, and T. J. R. Hughes. Conformal solid T-spline construction from boundary T-spline representations. *Computational Mechanics*, 51:1051–1059, 2013.

[134] Z. Zhang, J. Xu, B. Hong, and X. Chen. The effects of 3D channel geometry on CTC passing pressure - towards deformability-based cancer cell separation. *Lab Chip*, 14:2576–2584, 2014.

[135] O. C. Zienkiewicz, R. L. Taylor, O. C. Zienkiewicz, and R. L. Taylor. *The finite element method*. McGraw-hill London, 1977.

Appendices

A Paper #1: "A NURBS-based immersed methodology for fluid-structure interaction"

Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 284 (2015) 943–970
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

A NURBS-based immersed methodology for fluid–structure interaction

Hugo Casquero*, Carles Bona-Casas, Hector Gomez

Departamento de Métodos Matemáticos, Universidade da Coruña, Campus de A Coruña, 15071, A Coruña, Spain

Available online 25 November 2014

Abstract

We introduce an isogeometric, immersed, and fully-implicit formulation for fluid–structure interaction (FSI). The method focuses on viscous incompressible flows and nonlinear hyperelastic incompressible solids, which are a common case in various fields, such as, for example, biomechanics. In our FSI method, we utilize an Eulerian mesh on the whole domain and a Lagrangian mesh on the solid domain. The Lagrangian mesh is arbitrarily located on top of the Eulerian mesh in a non-conforming fashion. Due to the formulation of our problem, based on the Immersed Finite Element Method (IFEM), we do not need mesh update or remeshing algorithms. The fluid–structure interface is the boundary of the Lagrangian mesh, but cuts arbitrarily the Eulerian mesh. The generalized- α method is used for time discretization and NURBS-based isogeometric analysis is employed for the spatial discretization on both fluid and solid domains. The information transfer between the two meshes is carried out using the NURBS functions, which avoids the use of the so-called discretized delta functions. The higher order and especially the higher continuity of NURBS functions allow us to deal with severe mesh distortion in the Lagrangian mesh in comparison with classical C^0 linear piecewise functions as we prove numerically. Our numerical solutions attain good agreement with theoretical data for free-falling objects in two and three dimensions, which confirms the feasibility of our methodology.

Keywords: Fluid–structure interaction; Isogeometric analysis; Immersed methods; NURBS; Variational multiscale; Collocation methods

1. Introduction

In the past few decades, numerous research efforts have been directed to the development of modeling and simulation techniques for fluid–structure interaction (FSI) problems. See, for example [1–11]. An efficient, accurate, and robust computational technique is essential in studying complicated FSI problems, which are ubiquitous in all fields of engineering. Body-fitted approaches [1] have been widely utilized in FSI problems and are typically based upon the arbitrary Lagrangian–Eulerian (ALE) description [12–15]. The ALE description is used to track the fluid–solid interface properly, which is one of the main challenges in a FSI technique. These approaches are usually

* Corresponding author.
E-mail address: hugo.casquero@udc.es (H. Casquero).

http://dx.doi.org/10.1016/j.cma.2014.10.055
0045-7825/

very accurate, but mesh update or remeshing algorithms are needed, which are time consuming and can lead to inaccuracies in the variables that are projected onto the mesh in some cases.

A different type of FSI methodology was introduced by Peskin in the 1970s [16]. Peskin developed the immersed boundary (IB) method, which may be applied to solids that take the form of a fiber network and are completely submerged into the fluid. In Peskin's method, a uniform, fixed Eulerian mesh spans the whole domain and the submerged solid is represented by a fiber network which is arbitrarily located on top of the Eulerian mesh. A discretized Dirac delta function with $\mathcal{C}^1$ continuity is employed to transfer information between the solid and the Eulerian mesh. This information transfer represents the interaction between the fluid and the solid, which is accounted for by distributing interaction forces in the fluid and interpolating the solid velocities from the fluid velocities. In this way, the action of the solid on the fluid is represented by a forcing function. Thus, on the Eulerian mesh only the forced Navier–Stokes equations need to be solved and no mesh-update algorithms are necessary. The IB method has been applied to a large variety of problems such as the design of prosthetic cardiac valves [17], wave propagation in cochlea [18], biofilm processes [19], and swimming motions of marine worms [20].

In the 2000s, the IB method was enhanced and extended by Liu and his coworkers giving rise to a new algorithm called immersed finite element method (IFEM) [21]. One of the major limitations of the IB method is that the solid carries mass, but does not occupy volume. This limitation is circumvented in the IFEM. The IFEM uses finite elements for both the fluid and the solid. In origin, the IFEM also employed a discretized delta function for the information transfer between meshes, but it was removed later in [22] and the finite element shape functions were used instead. The IFEM has been used to solve several problems of biomechanics such as stent deployment [23], cell migration, and cell–substrate adhesion [6] and hemodynamics [24,25]. We note, however, that as far as we are aware, the IFEM has not been applied to flows with very large Reynolds number and these problems could be difficult to tackle by the IFEM because the boundary layer at the fluid–structure interface must be resolved by the Eulerian mesh.

We believe that the IFEM was a very significant development which opened the door to the solution of FSI problems that could not have been computed otherwise. Nevertheless, the generality of the IFEM provides opportunities for different interpretations, and opens the way to potential improvements. In particular, some of the limitations of the IFEM seem to emanate from its underlying technology, namely the finite element method. For instance, higher-order robustness and the possibility of handling severe mesh distortions in the solid domain seem to be areas in which there is opportunity for improvement. Our motivation here is to address these points by extending the IFEM to the isogeometric analysis (IGA) realm. Isogeometric analysis was created to fill the gap between computer aided design (CAD) and simulation [26]. The main idea of IGA is to utilize the functions that are commonly used in CAD as shape functions in analysis. Up to now, nonuniform rational B-splines (NURBS) have been the functions most widely employed in IGA. If we compare NURBS with Lagrange functions, which are typically used in the finite element method, NURBS have several advantages due to their higher continuity between elements and some of them are the following: First, spatial derivatives of second or higher order may be computed easily, which permits straightforward solution to higher-order partial differential equations [27–35]. Second, NURBS can handle severe mesh distortion in discretizations of solid mechanics as shown in [36]. Also, NURBS improve the accuracy on a per-degree-of-freedom basis as shown in several applications [37,38]. IGA has been successfully applied to a large number of fields including body-fitted FSI problems [15,39], fluid mechanics [40,41], phase-field models [27,42–46], structural vibrations [47,48], quasi-incompressible elasticity [49], shell modeling [50,51], contact problems [52–54], shape optimization [55], and electromagnetics [56]. A book about IGA has been written, detailing the methodology and its applications [57].

In this paper, we propose a new method to solve immersed FSI using isogeometric analysis both for the fluid and the solid. Therefore, we take advantage of all the aforementioned benefits of NURBS. The NURBS functions are also used to transfer the information between meshes. It is important to notice that this information transfer always reduces the efficiency and the accuracy of the overall algorithm, so we have designed our algorithms to minimize the amount of information transferred between meshes.

Our method only uses the *ab initio* laws of fluid and solid mechanics. Due to the applications we are interested in, the solids are treated as incompressible, nonlinear and hyperelastic, but other material models could be considered (for example, in [22] a similar algorithm is presented where compressible solids are considered, and in [58] the authors propose an immersed method capable of dealing with rigid and deformable solids). The fluid is governed by the Navier–Stokes equations of viscous incompressible flows. For the space discretization, we use the variational multiscale (VMS) method developed by Hughes et al. [59–61]. Previous studies on the IFEM employed explicit or

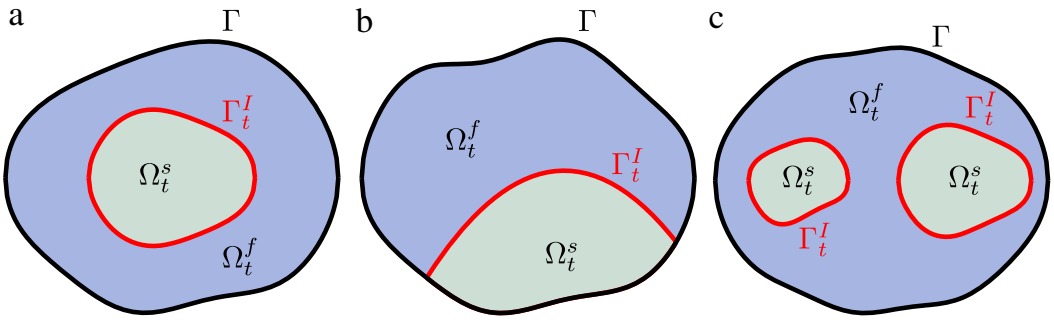


Fig. 1. Three possible configurations of the solid domain Ω_t^s (green), the fluid domain Ω_t^f (blue), and the fluid–solid interface Γ_t^I (red). Note that the solid domain is allowed to be non-connected (see snapshot (c)). The domain Ω is implicitly defined as the set enclosed by Γ. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

semi-implicit first-order accurate time integration schemes. Here, we propose a fully-implicit second-order accurate time integration algorithm based on the generalized-α method [62,63].

The outline of this paper is as follows. In Section 2 we present the mathematical formulation of the immersed FSI problem at the continuous level. We start describing the kinematics of the fluid–solid system. Then, we focus on the strong form and the weak form of the problem. Section 3 describes the discretization process. We employ a semi-discrete formulation and make use of the VMS method which includes additional terms in comparison with classical stabilized methods [64]. The interaction between the fluid and the solid is described in detail in this section. We also provide the main ideas to implement the method. Section 4 displays several numerical examples in 2D and 3D. The first example is a free-falling cylinder in 2D and our numerical results are checked against an exact solution when it is available. The second example is a free falling sphere in 3D and the numerical results are again compared with a theoretical solution. Our results show good agreement with the theoretical data in both cases, which confirms the viability of the approach. In a third example, we consider a parallelogram submerged in a flow as a test problem to show that higher-order and higher-continuity NURBS functions allow to deal with larger deformations in the Lagrangian mesh in comparison with classical $\mathcal{C}^0$ linear piecewise functions. Section 4 finishes with a simulation of six hollow cylinders submerged in a flow in order to exhibit the capability of the method to handle problems with several immersed solids. Finally, in Section 5, some concluding remarks will be drawn.

2. Formulation of the immersed problem at the continuous level

Let $\Omega \subset \mathbb{R}^d$ be an open set, where d is the number of spatial dimensions. The spatial domain Ω encloses the entirety of our fluid–solid system. Although using standard techniques for problems on moving domains (e.g., ALE or space–time methods; see [1]), our methodology can be applied to the case in which Ω depends upon time, all the numerical examples presented in this paper are posed on a fixed domain Ω, so we will restrict our presentation to this case from the outset. In what follows, $\Omega_t^f \subset \Omega$ denotes the open subset of Ω occupied by the fluid at time t. For simplicity, we will call Ω_t^f "fluid domain". The solid domain, which is also an open subset of Ω, will be denoted Ω_t^s. The solid and fluid domains define a partition of Ω, in such a way that $\overline{\Omega} = \overline{\Omega_t^f \cup \Omega_t^s}$. Note that although both the fluid and solid domain depend on time, Ω has been assumed to be time independent, as this is the relevant case for all the numerical examples in this paper which involve free-falling objects and particulate flows. The fluid and solid domains are not allowed to overlap ($\Omega_t^f \cap \Omega_t^s = \emptyset$, where $\emptyset$ is the empty set), but they meet at the solid–fluid interface that we call Γ_t^I ($\overline{\Omega_t^f} \cap \overline{\Omega_t^s} = \Gamma_t^I$). The boundaries of Ω, Ω_t^f and Ω_t^s are denoted by Γ, Γ_t^f and Γ_t^s, respectively, and are assumed to have a well-defined unit outward normal. For future reference, we denote these normals by $\boldsymbol{n}$, $\boldsymbol{n}^f$ and $\boldsymbol{n}^s$, where the subscript t is omitted for notational simplicity. Fig. 1 shows three possible configurations. Note that the set Ω_t^s is allowed to be non-connected (see Fig. 1(c)).

2.1. Kinematics

To introduce the basic notation about motions and configurations, let us define a reference configuration Ω_0^s for our solid body. In principle, a reference configuration for the fluid domain could also be defined, but we will not make

use of it. We consider the mapping $\boldsymbol{\varphi} : \Omega_0^s \times (0, T) \mapsto \mathbb{R}^d$, where $(0, T)$ is the time interval of interest.[1] We assume that $\boldsymbol{\varphi}$ is sufficiently smooth, orientation preserving and invertible [65]. Points $\boldsymbol{X}$ in Ω_0^s are called material points or particles, while points in $\mathbb{R}^d$ are denoted $\boldsymbol{x}$ and are called spatial points. To define the mapping $\boldsymbol{\varphi}$ we make use of the displacement field $\overline{\boldsymbol{u}}^s : \Omega_0^s \times (0, T) \mapsto \mathbb{R}^d$. In particular,

$$\boldsymbol{\varphi}(\boldsymbol{X}, t) = \boldsymbol{X} + \overline{\boldsymbol{u}}^s(\boldsymbol{X}, t). \tag{1}$$

For convenience, we assume that $\Omega_t^s = \boldsymbol{\varphi}(\Omega_0^s, t)$ and, thus we associate Ω_0^s with the undeformed configuration of the solid. The deformation gradient $\boldsymbol{F} : \Omega_0^s \times (0, T) \mapsto \mathbb{R}^{d\times d}$ is defined as

$$\boldsymbol{F} = \nabla_{\boldsymbol{X}} \boldsymbol{\varphi} = \boldsymbol{I} + \nabla_{\boldsymbol{X}} \overline{\boldsymbol{u}}^s \tag{2}$$

where the subscript $\boldsymbol{X}$ in $\nabla_{\boldsymbol{X}}$ emphasizes that the gradient applies to the variables $\boldsymbol{X}$. In (2), $\boldsymbol{I}$ denotes the identity tensor in $\mathbb{R}^{d\times d}$. Let us also define the Cauchy–Green deformation tensor $\boldsymbol{C} : \Omega_0^s \times (0, T) \mapsto \mathbb{R}^{d\times d}_{\text{sym}}$ as

$$\boldsymbol{C} = \boldsymbol{F}\boldsymbol{F}^T,$$

where $\mathbb{R}^{d\times d}_{\text{sym}}$ denotes the space of symmetric tensors in $\mathbb{R}^{d\times d}$, and $\boldsymbol{F}^T$ represents the transpose of $\boldsymbol{F}$. We will also make use of the Green–Lagrange strain tensor $\boldsymbol{E} : \Omega_0^s \times (0, T) \mapsto \mathbb{R}^{d\times d}_{\text{sym}}$ defined as

$$\boldsymbol{E} = \frac{1}{2}(\boldsymbol{C} - \boldsymbol{I}). \tag{3}$$

Finally, we define the velocity $\overline{\boldsymbol{v}}^s : \Omega_0^s \times (0, T) \mapsto \mathbb{R}^d$, and the acceleration $\overline{\boldsymbol{a}}^s : \Omega_0^s \times (0, T) \mapsto \mathbb{R}^d$ of a material point as

$$\overline{\boldsymbol{v}}^s(\boldsymbol{X}, t) = \frac{\partial \overline{\boldsymbol{u}}^s(\boldsymbol{X}, t)}{\partial t} \tag{4}$$

$$\overline{\boldsymbol{a}}^s(\boldsymbol{X}, t) = \frac{\partial \overline{\boldsymbol{v}}^s(\boldsymbol{X}, t)}{\partial t} = \frac{\partial^2 \overline{\boldsymbol{u}}^s(\boldsymbol{X}, t)}{\partial t^2}. \tag{5}$$

Associated to the function $\overline{\boldsymbol{v}}^s$, we define $\boldsymbol{v}^s : \Omega_t^s \times (0, T) \mapsto \mathbb{R}^d$ as

$$\overline{\boldsymbol{v}}^s(\boldsymbol{X}, t) = \boldsymbol{v}^s(\boldsymbol{\varphi}(\boldsymbol{X}, t), t) \quad \text{for all } \boldsymbol{X} \in \Omega_0^s,\ t \in [0, T]. \tag{6}$$

Note that although $\overline{\boldsymbol{v}}^s$ and $\boldsymbol{v}^s$ represent the same physical quantity, namely the solid velocity, they are different functions, and thus, we use different notation for them. In what follows, we will call $\overline{\boldsymbol{v}}^s$ Lagrangian or material velocity and $\boldsymbol{v}^s$ Eulerian or spatial velocity. In general, we will use a bar for Lagrangian functions whenever there is possibility of confusion. Applying the chain rule to (6), we obtain

$$\overline{\boldsymbol{a}}^s(\boldsymbol{X}, t) = \frac{\partial \overline{\boldsymbol{v}}^s}{\partial t}(\boldsymbol{X}, t) = \frac{\partial \boldsymbol{v}^s}{\partial t}(\boldsymbol{x}, t) + \boldsymbol{v}^s(\boldsymbol{x}, t) \cdot \nabla_{\boldsymbol{x}} \boldsymbol{v}^s(\boldsymbol{x}, t) \quad \text{if } \boldsymbol{x} = \boldsymbol{\varphi}(\boldsymbol{X}, t). \tag{7}$$

In what follows, we will also make use of the standard notation

$$\dot{\boldsymbol{\alpha}} = \frac{\mathrm{d}\boldsymbol{\alpha}}{\mathrm{d}t} = \frac{\partial \boldsymbol{\alpha}}{\partial t} + \boldsymbol{v} \cdot \nabla_{\boldsymbol{x}} \boldsymbol{\alpha} \tag{8}$$

for a generic Eulerian function $\boldsymbol{\alpha}$. For future reference, we also state the relation between the gradient of $\boldsymbol{\alpha}$ and the gradient of its Lagrangian counterpart $\overline{\boldsymbol{\alpha}}$, which is given by

$$\nabla_{\boldsymbol{x}} \boldsymbol{\alpha}(\boldsymbol{x}, t) = \nabla_{\boldsymbol{X}} \overline{\boldsymbol{\alpha}}(\boldsymbol{X}, t) \boldsymbol{F}^{-1}(\boldsymbol{X}, t) \quad \text{if } \boldsymbol{x} = \boldsymbol{\varphi}(\boldsymbol{X}, t). \tag{9}$$

[1] In a situation like that represented in Fig. 1(c), in which we have several solids immersed into the fluid, we would have to define one mapping $\boldsymbol{\varphi}$ associated to each of the solids.

2.2. Strong form of the problem

To derive the strong form of the governing equations, we assume that both fluid and solid densities are constant, which implies incompressibility of both fluid and solid phases. As a consequence, the entire fluid–solid system is also incompressible, even if the density may be non-homogeneous in space when fluid and solid densities are different (for a thorough study of incompressible media with variable density, see [66]). Under these considerations, and assuming that gravity is the only external force acting on the system, the governing equations may be written in Eulerian coordinates as

$$\rho^f \frac{\mathrm{d}\boldsymbol{v}^f}{\mathrm{d}t} = \nabla_{\boldsymbol{x}} \cdot \boldsymbol{\sigma}^f + \rho^f \boldsymbol{g} \quad \text{in } \Omega_t^f \times (0,T) \tag{10.1}$$

$$\nabla_{\boldsymbol{x}} \cdot \boldsymbol{v}^f = 0 \quad \text{in } \Omega_t^f \times (0,T) \tag{10.2}$$

$$\rho^s \frac{\mathrm{d}\boldsymbol{v}^s}{\mathrm{d}t} = \nabla_{\boldsymbol{x}} \cdot \boldsymbol{\sigma}^s + \rho^s \boldsymbol{g} \quad \text{in } \Omega_t^s \times (0,T) \tag{10.3}$$

$$\nabla_{\boldsymbol{x}} \cdot \boldsymbol{v}^s = 0 \quad \text{in } \Omega_t^s \times (0,T) \tag{10.4}$$

$$\boldsymbol{v}^f = \boldsymbol{v}^s \quad \text{on } \Gamma_t^I \times (0,T) \tag{10.5}$$

$$\boldsymbol{\sigma}^f \boldsymbol{n}^f = -\boldsymbol{\sigma}^s \boldsymbol{n}^s \quad \text{on } \Gamma_t^I \times (0,T) \tag{10.6}$$

where ρ^f, $\boldsymbol{v}^f$ and $\boldsymbol{\sigma}^f$ denote the fluid density, velocity and Cauchy stress tensor, respectively, while ρ^s, $\boldsymbol{v}^s$ and $\boldsymbol{\sigma}^s$ denote the same quantities for the solid. Eq. (10.5) represents a kinematic constraint that equates the fluid velocity with that of the solid at the interface, while (10.6) ensures correct transmission of forces at the interface. Eqs. (10.1) and (10.3) are associated to the unknowns $\boldsymbol{v}^f$ and $\boldsymbol{v}^s$, while constraints (10.2) and (10.4) are associated to Lagrange multipliers represented by pressure fields p^f and p^s in the stress tensors $\boldsymbol{\sigma}^f$ and $\boldsymbol{\sigma}^s$. We also note that in order to get a well-posed initial/boundary-value problem, we will need to impose initial conditions on Ω_t^f and boundary conditions on $\Gamma \cap \Gamma_t^f$ for a fluid mechanics problem along with initial conditions on Ω_t^s and boundary conditions on $\Gamma \cap \Gamma_t^s$ for a solid mechanics problem.

2.2.1. Global velocity and pressure fields

The key idea of the method is to define global functions $\boldsymbol{v} : \Omega \times (0,T) \mapsto \mathbb{R}^d$, $p : \Omega \times (0,T) \mapsto \mathbb{R}$, such that

$$\boldsymbol{v} = \begin{cases} \boldsymbol{v}^f & \text{on } \Omega_t^f \times (0,T) \\ \boldsymbol{v}^s & \text{on } \Omega_t^s \times (0,T); \end{cases} \qquad p = \begin{cases} p^f & \text{on } \Omega_t^f \times (0,T) \\ p^s & \text{on } \Omega_t^s \times (0,T). \end{cases} \tag{11}$$

The function $\boldsymbol{v}$ is globally continuous due to the kinematic constraint (10.5), but the pressure field could be discontinuous in the limit case of a solid body with zero measure (see [67,68]). Using the function $\boldsymbol{v}$, the strong form of the governing equations in Eulerian coordinates (10) may be rewritten as

$$\rho^f \frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}t} = \nabla_{\boldsymbol{x}} \cdot \boldsymbol{\sigma}^f + \rho^f \boldsymbol{g} + \boldsymbol{\mathcal{F}} \quad \text{in } \Omega \times (0,T) \tag{12.1}$$

$$\nabla_{\boldsymbol{x}} \cdot \boldsymbol{v} = 0 \quad \text{in } \Omega \times (0,T) \tag{12.2}$$

$$\boldsymbol{\sigma}^f \boldsymbol{n}^f = -\boldsymbol{\sigma}^s \boldsymbol{n}^s \quad \text{on } \Gamma_t^I \times (0,T) \tag{12.3}$$

where $\boldsymbol{\mathcal{F}}$ will be, in principle, a discontinuous function and it is defined in such a way that Eq. (12.1) is equivalent to Eqs. (10.1) and (10.3)

$$\boldsymbol{\mathcal{F}} = \begin{cases} 0, & \boldsymbol{x} \in \Omega_t^f \\ \left(\rho^f - \rho^s\right)(\dot{\boldsymbol{v}} - \boldsymbol{g}) + \nabla_{\boldsymbol{x}} \cdot \left(\boldsymbol{\sigma}^s - \boldsymbol{\sigma}^f\right), & \boldsymbol{x} \in \Omega_t^s. \end{cases} \tag{13}$$

Note that Eq. (10.5) is automatically satisfied due to the continuity of $\boldsymbol{v}$ across the fluid–solid interface, and thus, it is no longer necessary. In principle, Eq. (12.3) requires tracking the fluid–solid interface, but as we will show later, our variational formulation imposes this constraint naturally, and no algorithmical treatment will be needed.

2.2.2. Constitutive equations

To completely define Eqs. (12) and (13) we need to determine the stress tensors $\boldsymbol{\sigma}^f$ ans $\boldsymbol{\sigma}^s$. We will assume the fluid to be Newtonian and incompressible. Therefore, $\boldsymbol{\sigma}^f : \Omega \times (0, T) \mapsto \mathbb{R}^{d\times d}$

$$\boldsymbol{\sigma}^f = -p\boldsymbol{I} + 2\mu\nabla_{\boldsymbol{x}}^{\text{sym}}\boldsymbol{v} \tag{14}$$

where $\mu > 0$ is the dynamic viscosity and $\nabla_{\boldsymbol{x}}^{\text{sym}}\boldsymbol{v} = (\nabla_{\boldsymbol{x}}\boldsymbol{v} + \nabla_{\boldsymbol{x}}\boldsymbol{v}^T)/2$. Note that we have extended the definition of $\boldsymbol{\sigma}^f$ to the entire fluid–solid system, even if it does not have a clear physical meaning on Ω_t^s.

The Cauchy stress tensor of the solid is given by the expression

$$\boldsymbol{\sigma}^s = -p\boldsymbol{I} + \boldsymbol{F}\boldsymbol{S}\boldsymbol{F}^T/J \tag{15}$$

where $J = \det(\boldsymbol{F})$ and $\boldsymbol{S}$ is the second Piola–Kirchhoff stress tensor. We will consider hyperelastic materials, assuming the existence of a stored elastic-energy function per unit volume of the undeformed configuration ψ. Following the standard additive decomposition of ψ (see, e.g., [69]) into a volumetric part depending only on J and an isochoric part, we obtain

$$\psi(J, \boldsymbol{C}) = \psi^{\text{dil}}(J) + \psi^{\text{iso}}(\boldsymbol{C}). \tag{16}$$

Here, we will use the Neo-Hookean materials with dilatation penalty proposed in [69], which are defined as,

$$\psi^{\text{dil}}(J) = \frac{1}{2}\kappa^s\left(\frac{1}{2}(J^2 - 1) - \ln J\right) \tag{17}$$

$$\psi^{\text{iso}}(J^{-2/d}\boldsymbol{C}) = \frac{1}{2}\mu^s\left(J^{-2/d}\text{tr}(\boldsymbol{C}) - d\right) \tag{18}$$

where $\text{tr}(\cdot)$ denotes the trace operator, κ^s is the bulk modulus and μ^s is the shear modulus. For an incompressible material $\psi^{\text{dil}}(J) = 0$, and the second stress tensor is computed as

$$\boldsymbol{S} = \frac{\partial\psi^{\text{iso}}}{\partial\boldsymbol{E}}. \tag{19}$$

To compute the stress tensor $\boldsymbol{S}$ we need the Lagrangian displacement of the solid $\overline{\boldsymbol{u}}^s(\boldsymbol{X}, t)$, which will be obtained from Eq. (4), repeated here for completeness

$$\frac{\partial\overline{\boldsymbol{u}}^s}{\partial t}(\boldsymbol{X}, t) = \boldsymbol{v}(\boldsymbol{x}, t) \quad \text{if } \boldsymbol{x} = \boldsymbol{\varphi}(\boldsymbol{X}, t). \tag{20}$$

Eq. (20) holds for all $\boldsymbol{X} \in \Omega_0^s$ and for all $t \in [0, T]$.

2.2.3. Initial and boundary conditions

As initial conditions, we set an initial velocity for the entire fluid–solid system. Since the solid displacements are also necessary to evaluate the solid stress tensor, we need to specify also initial displacements on Ω_0^s. Thus, our initial conditions are given by

$$\boldsymbol{v}(\boldsymbol{x}, 0) = \boldsymbol{v}_I(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega \tag{21}$$

$$\overline{\boldsymbol{u}}^s(\boldsymbol{X}, 0) = 0, \quad \boldsymbol{X} \in \Omega_0^s. \tag{22}$$

Note that Eq. (22) associates the reference configuration Ω_0^s with the undeformed configuration.

To define suitable boundary conditions we note that the boundary Γ can be subdivided into non-overlapping sets $\mathcal{B}^s = \Gamma \cap \Gamma_t^s$ and $\mathcal{B}^f = \Gamma \cap \Gamma_t^f$ such that $\Gamma = \overline{\mathcal{B}^s \cup \mathcal{B}^f}$ and $\emptyset = \mathcal{B}^s \cap \mathcal{B}^f$. Note, that in general $\mathcal{B}^s \neq \Gamma_t^s$, because part of Γ_t^s may not belong to Γ (see an example in Fig. 1(b)). To define a well-posed problem we need to specify standard boundary conditions for a fluid- and solid-mechanics problem on $\mathcal{B}^s$ and $\mathcal{B}^f$, respectively. Since we have limited ourselves to the case in which Ω is fixed in time, we only consider homogeneous displacement boundary conditions on $\mathcal{B}^s$. On $\mathcal{B}^f$, however, we allow for more general boundary conditions and we actually split $\mathcal{B}^f$ as $\mathcal{B}^f = \overline{\mathcal{B}_D^f \cup \mathcal{B}_N^f}$,

with $\mathcal{B}_D^f \cap \mathcal{B}_N^f = \emptyset$. On $\mathcal{B}_D^f$ we impose a given velocity $\boldsymbol{v}_B$ whereas on $\mathcal{B}_N^f$ we prescribe a traction vector $\boldsymbol{h}$. Thus, our boundary conditions are defined as

$$\boldsymbol{v} = \boldsymbol{v}_B \quad \text{on } \mathcal{B}_D^f \tag{23.1}$$

$$\boldsymbol{\sigma}^f \boldsymbol{n}^f = \boldsymbol{h} \quad \text{on } \mathcal{B}_N^f \tag{23.2}$$

$$\overline{\boldsymbol{u}}^s = 0 \quad \text{on } \mathcal{B}^s. \tag{23.3}$$

2.3. Weak form

Let $\mathcal{S}_v$ and $\mathcal{S}_p$ denote trial solution spaces for velocity and pressure, defined as follows

$$\mathcal{S}_v = \left\{ \boldsymbol{v} | \boldsymbol{v}(\cdot, t) \in \left(\mathcal{H}^1(\Omega)\right)^d,\ \boldsymbol{v} = \boldsymbol{v}_B \text{ on } \mathcal{B}_D^f,\ \boldsymbol{v} = 0 \text{ on } \mathcal{B}^s \right\} \tag{24.1}$$

$$\mathcal{S}_p = \left\{ p | p(\cdot, t) \in \mathcal{L}^2(\Omega),\ \int_\Omega p \mathrm{d}\Omega = 0 \text{ if } \mathcal{B}_N^f = \emptyset \right\}. \tag{24.2}$$

Here, $\mathcal{L}^2(\Omega)$ denotes the space of scalar-valued functions that are square-integrable on Ω, while $\left(\mathcal{H}^1(\Omega)\right)^d$ denotes the space of square-integrable $\mathbb{R}^d$-valued functions with square-integrable derivatives on Ω. Analogously, we define weighting function spaces $\mathcal{V}_v$, $\mathcal{V}_p$ for the momentum and continuity equations as follows

$$\mathcal{V}_v = \left\{ \boldsymbol{w} | \boldsymbol{w}(\cdot) \in \left(\mathcal{H}^1(\Omega)\right)^d,\ \boldsymbol{w} = 0 \text{ on } \mathcal{B}_D^f \cup \mathcal{B}^s \right\} \tag{25.1}$$

$$\mathcal{V}_p = \left\{ q | q(\cdot) \in \mathcal{L}^2(\Omega) \right\}. \tag{25.2}$$

To derive a weak form of our FSI problem, we begin by multiplying Eq. (12.1) with a weight function $\boldsymbol{w} \in \mathcal{V}_v$. Then, we integrate over the domain Ω, and use integration by parts to obtain

$$\left(\boldsymbol{w}, \rho^f(\dot{\boldsymbol{v}} - \boldsymbol{g})\right)_\Omega + \left(\nabla_{\boldsymbol{x}} \boldsymbol{w}, \boldsymbol{\sigma}^f\right)_\Omega - (\boldsymbol{w}, \boldsymbol{h})_{\mathcal{B}_N^f} - (\boldsymbol{w}, \boldsymbol{\mathcal{F}})_{\Omega_t^s} = 0 \tag{26}$$

where $(\cdot,\cdot)_\Omega$ denotes the $\mathcal{L}^2$ inner product over the domain Ω. To derive (26) we have used the fact that $\boldsymbol{w} = 0$ on $\mathcal{B}_D^f \cup \mathcal{B}^s$ and we have replaced $\boldsymbol{\sigma}^f \boldsymbol{n}^f$ with $\boldsymbol{h}$ on the third term of the left hand side. Let us focus now on the last term of the left hand side of (26)

$$\begin{aligned}
(\boldsymbol{w}, \boldsymbol{\mathcal{F}})_{\Omega_t^s} &= \left(\boldsymbol{w}, (\rho^f - \rho^s)(\dot{\boldsymbol{v}} - \boldsymbol{g})\right)_{\Omega_t^s} + \left(\boldsymbol{w}, \nabla_{\boldsymbol{x}} \cdot (\boldsymbol{\sigma}^s - \boldsymbol{\sigma}^f)\right)_{\Omega_t^s} \\
&= \left(\boldsymbol{w}, (\rho^f - \rho^s)(\dot{\boldsymbol{v}} - \boldsymbol{g})\right)_{\Omega_t^s} + \left(\nabla_{\boldsymbol{x}} \boldsymbol{w}, \boldsymbol{\sigma}^f - \boldsymbol{\sigma}^s\right)_{\Omega_t^s} \\
&\quad + \left(\boldsymbol{w}, (\boldsymbol{\sigma}^s - \boldsymbol{\sigma}^f)\boldsymbol{n}^s\right)_{\Gamma_t^I} + \left(\boldsymbol{w}, (\boldsymbol{\sigma}^s - \boldsymbol{\sigma}^f)\boldsymbol{n}^s\right)_{\mathcal{B}^s} \\
&= \left(\boldsymbol{w}, (\rho^f - \rho^s)(\dot{\boldsymbol{v}} - \boldsymbol{g})\right)_{\Omega_t^s} + \left(\nabla_{\boldsymbol{x}} \boldsymbol{w}, \boldsymbol{\sigma}^f - \boldsymbol{\sigma}^s\right)_{\Omega_t^s}
\end{aligned} \tag{27}$$

where we have used the fact that $\Gamma_t^s = \overline{\Gamma_t^I \cup \mathcal{B}^s}$. We have also used Eqs. (10.6), $\boldsymbol{n}^s = -\boldsymbol{n}^f$, and (25.1) to annihilate the boundary integrals. If we now pull back the integrals in Eq. (27) to the reference configuration Ω_0^s making use of the mapping $\boldsymbol{\varphi}$, and we use expression (7), we obtain

$$\begin{aligned}
(\boldsymbol{w}(\boldsymbol{x}), \boldsymbol{\mathcal{F}}(\boldsymbol{x}, t))_{\Omega_t^s} &= \left(\boldsymbol{w}(\boldsymbol{\varphi}(\boldsymbol{X}, t)), (\rho^f - \rho^s)\left(\frac{\partial \boldsymbol{v}}{\partial t}(\boldsymbol{\varphi}(\boldsymbol{X}, t), t) - \boldsymbol{g}\right) J(\boldsymbol{X}, t)\right)_{\Omega_0^s} \\
&\quad + \left(\boldsymbol{w}(\boldsymbol{\varphi}(\boldsymbol{X}, t)), (\rho^f - \rho^s)\boldsymbol{v}(\boldsymbol{\varphi}(\boldsymbol{X}, t), t) \cdot \nabla_{\boldsymbol{x}} \boldsymbol{v}(\boldsymbol{\varphi}(\boldsymbol{X}, t), t) J(\boldsymbol{X}, t)\right)_{\Omega_0^s} \\
&\quad + \left(\nabla_{\boldsymbol{x}} \boldsymbol{w}(\boldsymbol{\varphi}(\boldsymbol{X}, t)), \left[\boldsymbol{\sigma}^f(\boldsymbol{\varphi}(\boldsymbol{X}, t), t) - \boldsymbol{\sigma}^s(\boldsymbol{\varphi}(\boldsymbol{X}, t), t)\right] J(\boldsymbol{X}, t)\right)_{\Omega_0^s}.
\end{aligned} \tag{28}$$

Note that we may call the right hand side of (28) a mixed Lagrangian–Eulerian form because we are integrating over the reference configuration, but all derivatives are taken with respect to physical space variables. As we will see next, this is an important aspect of our algorithm and one that will permit us significant computational savings, and perhaps also increased accuracy (see Remark 2 in Sections 3.1 and 4.1). Eq. (28) completes the derivation of the weak form of (12.1). The weak form of Eq. (12.2) can be derived by multiplying the equation with a function $q \in \mathcal{V}_p$, and integrating over Ω to obtain

$$(q, \nabla_{\boldsymbol{x}} \cdot \boldsymbol{v})_{\Omega} = 0. \tag{29}$$

Now, we combine Eqs. (26), (28) and (29), dividing everywhere through the fluid density and renaming the p as the pressure divided by ρ^f. We also use the classical notation $\nu = \mu/\rho^f$. Under these circumstances, our weak form may be written as

$$B^{ED}(\{\boldsymbol{w}, q\}, \{\boldsymbol{v}, p\}) - L^{ED}(\boldsymbol{w}) + B^{LD}(\boldsymbol{w}, \boldsymbol{v}) - L^{LD}(\boldsymbol{w}) = 0 \tag{30}$$

where

$$\begin{aligned} B^{ED}(\{\boldsymbol{w}, q\}, \{\boldsymbol{v}, p\}) \;=\; & \left(\boldsymbol{w}(\boldsymbol{x}), \frac{\partial \boldsymbol{v}}{\partial t}(\boldsymbol{x}, t) + \boldsymbol{v}(\boldsymbol{x}, t) \cdot \nabla_{\boldsymbol{x}} \boldsymbol{v}(\boldsymbol{x}, t)\right)_{\Omega} - (\nabla_{\boldsymbol{x}} \cdot \boldsymbol{w}(\boldsymbol{x}), p(\boldsymbol{x}))_{\Omega} \\ & + \left(\nabla_{\boldsymbol{x}}^{\mathrm{sym}} \boldsymbol{w}(\boldsymbol{x}), 2\nu \nabla_{\boldsymbol{x}}^{\mathrm{sym}} \boldsymbol{v}(\boldsymbol{x}, t)\right)_{\Omega} + (q(\boldsymbol{x}), \nabla_{\boldsymbol{x}} \cdot \boldsymbol{v}(\boldsymbol{x}, t))_{\Omega} \end{aligned} \tag{31}$$

$$L^{ED}(\boldsymbol{w}) = (\boldsymbol{w}(\boldsymbol{x}), \boldsymbol{g})_{\Omega} + \left(\boldsymbol{w}(\boldsymbol{x}), \boldsymbol{h}(\boldsymbol{x})/\rho^f\right)_{\mathcal{B}_N^f} \tag{32}$$

$$\begin{aligned} B^{LD}(\boldsymbol{w}, \boldsymbol{v}) \;=\; & -\left(\boldsymbol{w}(\boldsymbol{\varphi}(\boldsymbol{X}, t)), \left(1 - \frac{\rho^s}{\rho^f}\right) \frac{\partial \boldsymbol{v}(\boldsymbol{\varphi}(\boldsymbol{X}, t), t)}{\partial t} J(\boldsymbol{X}, t)\right)_{\Omega_0^s} \\ & - \left(\boldsymbol{w}(\boldsymbol{\varphi}(\boldsymbol{X}, t)), \left(1 - \frac{\rho^s}{\rho^f}\right) \boldsymbol{v}(\boldsymbol{\varphi}(\boldsymbol{X}, t), t) \cdot \nabla_{\boldsymbol{x}} \boldsymbol{v}(\boldsymbol{\varphi}(\boldsymbol{X}, t), t) J(\boldsymbol{X}, t)\right)_{\Omega_0^s} \\ & - \left(\nabla_{\boldsymbol{x}}^{\mathrm{sym}} \boldsymbol{w}(\boldsymbol{\varphi}(\boldsymbol{X}, t)), 2\nu \nabla_{\boldsymbol{x}}^{\mathrm{sym}} \boldsymbol{v}(\boldsymbol{\varphi}(\boldsymbol{X}, t), t) J(\boldsymbol{X}, t)\right)_{\Omega_0^s} \\ & + \left(\nabla_{\boldsymbol{x}}^{\mathrm{sym}} \boldsymbol{w}(\boldsymbol{\varphi}(\boldsymbol{X}, t)), \frac{1}{\rho^f} \boldsymbol{F}(\boldsymbol{X}, t) \boldsymbol{S}(\boldsymbol{X}, t) \boldsymbol{F}^T(\boldsymbol{X}, t)\right)_{\Omega_0^s} \end{aligned} \tag{33}$$

$$L^{LD}(\boldsymbol{w}) = -\left(\boldsymbol{w}(\boldsymbol{\varphi}(\boldsymbol{X}, t)), \left(1 - \frac{\rho^s}{\rho^f}\right) \boldsymbol{g} J(\boldsymbol{X}, t)\right)_{\Omega_0^s}. \tag{34}$$

Note that $J(\boldsymbol{X}, t) = 1$ due to the incompressibility constraint, but we have carried it in our derivations for generality reasons. Clearly, to evaluate the operators B^{LD} and L^{LD} we need the displacement field on the solid $\overline{\boldsymbol{u}}^s$, which will be computed using (20). In an IGA or finite-element context, perhaps the most natural approach would be to derive a weak form of (20) to eventually approximate $\overline{\boldsymbol{u}}^s$ by a member of a finite element space. This is of course a viable approach, but we find more appealing (and, very likely, more efficient) the possibility of approximating directly the strong form of (20) using an isogeometric collocation-like approach [70–73]. Thus, we do not proceed further with Eq. (20) at this point.

2.4. Scale separation and variational multiscale method

We wish to propose an algorithm that is stable and accurate for arbitrary combinations of the velocity and pressure spaces, as well as for problems with low and high Reynolds numbers. Thus, we derive a variational multiscale formulation of our weak form (30), following [40]. To do so, a direct sum decomposition of the solution spaces $\mathcal{S}_v$ and $\mathcal{S}_p$ is established into coarse-scale ($\widetilde{\mathcal{S}}_v$ and $\widetilde{\mathcal{S}}_p$) and fine-scale subspaces ($\mathcal{S}'_v$ and $\mathcal{S}'_p$). Thus,

$$\mathcal{S}_v = \widetilde{\mathcal{S}}_v \oplus \mathcal{S}'_v \tag{35}$$

$$\mathcal{S}_p = \widetilde{\mathcal{S}}_p \oplus \mathcal{S}'_p. \tag{36}$$

Therefore, for all $\boldsymbol{v} \in \mathcal{S}_v$ and for all $p \in \mathcal{S}_p$ there exists a unique decomposition

$$\boldsymbol{v} = \widetilde{\boldsymbol{v}} + \boldsymbol{v}', \quad \widetilde{\boldsymbol{v}} \in \widetilde{\mathcal{S}}_v \text{ and } \boldsymbol{v}' \in \mathcal{S}'_v \tag{37}$$

$$p = \widetilde{p} + p', \quad \widetilde{p} \in \widetilde{\mathcal{S}}_p \text{ and } p' \in \mathcal{S}'_p. \tag{38}$$

This decomposition is also applied to the weighting functions $\boldsymbol{w}$ and q. If we substitute the splitting (37)–(38) into our weak form (30), and neglect the effect of the velocity fine scales on the operator B^{LD}, that is, $B^{LD}(\widetilde{\boldsymbol{w}}, \widetilde{\boldsymbol{v}} + \boldsymbol{v}') = B^{LD}(\widetilde{\boldsymbol{w}}, \widetilde{\boldsymbol{v}})$, we obtain

$$B^{ED}\left(\{\widetilde{\boldsymbol{w}}, \widetilde{q}\}, \{\widetilde{\boldsymbol{v}} + \boldsymbol{v}', \widetilde{p} + p'\}\right) - L^{ED}(\widetilde{\boldsymbol{w}}) + B^{LD}(\widetilde{\boldsymbol{w}}, \widetilde{\boldsymbol{v}}) - L^{LD}(\widetilde{\boldsymbol{w}}) = 0. \tag{39}$$

Following [40], we make the following assumptions

1. The velocity fine scales vanish on the boundary, that is, $\boldsymbol{v}' = 0$ on Γ.
2. $(\widetilde{\boldsymbol{w}}, \frac{\partial \boldsymbol{v}'}{\partial t}) = 0$. More sophisticated approaches using dynamic velocity fine scales have been devised [74], leading to enhanced accuracy. However, we decided to favor simplicity and neglected the dynamics of the velocity fine scales.
3. $(\nabla_{\boldsymbol{x}}^{\text{sym}} \widetilde{\boldsymbol{w}}, 2\nu \nabla_{\boldsymbol{x}}^{\text{sym}} \boldsymbol{v}') = 0$. Note that this term can be omitted selecting a projector that enforces the orthogonality of the coarse and fine scales in the semi-norm induced by this term as demonstrated in [61].
4. The flow is incompressible, that is, $\nabla_{\boldsymbol{x}} \cdot (\widetilde{\boldsymbol{v}} + \boldsymbol{v}') = 0$.

Under the above hypotheses, the operator B^{ED} in Eq. (39) can be expressed as

$$\begin{aligned} B^{ED}\left(\{\widetilde{\boldsymbol{w}}, \widetilde{q}\}, \{\widetilde{\boldsymbol{v}} + \boldsymbol{v}', \widetilde{p} + p'\}\right) = & \left(\widetilde{\boldsymbol{w}}, \frac{\partial \widetilde{\boldsymbol{v}}}{\partial t} + \widetilde{\boldsymbol{v}} \cdot \nabla_{\boldsymbol{x}} \widetilde{\boldsymbol{v}}\right)_{\Omega} - (\nabla_{\boldsymbol{x}} \cdot \widetilde{\boldsymbol{w}}, \widetilde{p})_{\Omega} + \left(\nabla_{\boldsymbol{x}}^{\text{sym}} \widetilde{\boldsymbol{w}}, 2\nu \nabla_{\boldsymbol{x}}^{\text{sym}} \widetilde{\boldsymbol{v}}\right)_{\Omega} \\ & + (\widetilde{q}, \nabla_{\boldsymbol{x}} \cdot \widetilde{\boldsymbol{v}})_{\Omega} + \left(\widetilde{\boldsymbol{w}}, \boldsymbol{v}' \cdot \nabla \widetilde{\boldsymbol{v}}\right)_{\Omega} \\ & - \left(\nabla_{\boldsymbol{x}} \widetilde{\boldsymbol{w}}, (\widetilde{\boldsymbol{v}} + \boldsymbol{v}') \otimes \boldsymbol{v}'\right)_{\Omega} - \left(\nabla_{\boldsymbol{x}} \cdot \widetilde{\boldsymbol{w}}, p'\right)_{\Omega} - \left(\nabla_{\boldsymbol{x}} \widetilde{q}, \boldsymbol{v}'\right)_{\Omega}. \end{aligned} \tag{40}$$

3. Formulation of the immersed problem at the discrete level

Here, we apply a semi-discrete formulation to the problem defined in Eqs. (39)–(40). Our space discretization is based on IGA. We also make use of an isogeometric collocation-like process to transmit information from the fluid to the solid. We discretize in time using a second-order accurate, and fully-implicit method based on generalized-α.

3.1. Space discretization

To discretize the weak form (39) we need to define conforming finite-dimensional trial solution spaces $\mathcal{S}_{\boldsymbol{v}}^h \subset \widetilde{\mathcal{S}}_{\boldsymbol{v}}$ and $\mathcal{S}_p^h \subset \widetilde{\mathcal{S}}_p$, where h is a mesh parameter. We also define finite-dimensional weighting function spaces $\mathcal{V}_{\boldsymbol{v}}^h \subset \widetilde{\mathcal{V}}_{\boldsymbol{v}}$ and $\mathcal{V}_p^h \subset \widetilde{\mathcal{V}}_p$. Following standard arguments, we will replace $\widetilde{\boldsymbol{v}}$ and $\widetilde{p}$ in Eq. (39) with $\boldsymbol{v}^h \in \mathcal{S}_{\boldsymbol{v}}^h$ and $p^h \in \mathcal{S}_p^h$. The weighting functions $\widetilde{\boldsymbol{w}}$ and $\widetilde{q}$ will also be replaced with $\boldsymbol{w}^h \in \mathcal{V}_{\boldsymbol{v}}^h$ and $q^h \in \mathcal{V}_p^h$. If we ignore boundary conditions at this stage, our discrete spaces verify $\mathcal{S}_v^h = \mathcal{V}_v^h$ and $\mathcal{S}_p^h = \mathcal{V}_p^h$. In our algorithm, all the discrete spaces are spanned by NURBS, using the concept of IGA (for an introduction to IGA, see [57] or [26]). Let us denote by $\{N_A(\boldsymbol{x})\}_{A=1}^{n_{ED}}$ a set of NURBS basis functions defined in physical space. These functions define the discrete spaces we introduced above, which have dimension n_{ED}, where the subscript ED emphasizes that these spaces are defined in Eulerian coordinates. In what follows, we will suppose that p denotes the degree of the NURBS basis functions. Assuming that Dirichlet boundary conditions will be strongly enforced on the discrete space at a later stage, we can write

$$\boldsymbol{v}^h(\boldsymbol{x}, t) = \sum_{A=1}^{n_{ED}} \boldsymbol{v}_A(t) N_A(\boldsymbol{x}); \qquad p^h(\boldsymbol{x}, t) = \sum_{A=1}^{n_{ED}} p_A(t) N_A(\boldsymbol{x}) \tag{41}$$

$$\boldsymbol{w}^h(\boldsymbol{x}) = \sum_{A=1}^{n_{ED}} \boldsymbol{w}_A N_A(\boldsymbol{x}); \qquad q^h(\boldsymbol{x}) = \sum_{A=1}^{n_{ED}} q_A N_A(\boldsymbol{x}). \tag{42}$$

Invoking the isoparametric concept, the basis functions $\{N_A(\boldsymbol{x})\}_{A=1}^{n_{ED}}$ are also used to define a computational mesh on Ω. This computational mesh is used to compute the integrals on Ω that appear on the weak form (39) and will be called Eulerian mesh. Under these considerations, the semi-discrete form of our problem may be obtained by replacing

$\widetilde{\boldsymbol{w}} \leftarrow \boldsymbol{w}^h, \widetilde{\boldsymbol{v}} \leftarrow \boldsymbol{v}^h, \widetilde{p} \leftarrow p^h, \widetilde{q} \leftarrow q^h$, and computing $\boldsymbol{v}'$ and p' as

$$\boldsymbol{v}' = -\tau_M \boldsymbol{r}_M \left(\boldsymbol{v}^h, p^h\right) \tag{43.1}$$

$$p' = -\tau_C r_C \left(\boldsymbol{v}^h\right) \tag{43.2}$$

where τ_M and τ_C are defined as in [40], and

$$\boldsymbol{r}_M \left(\boldsymbol{v}^h, p^h\right) = \frac{\partial \boldsymbol{v}^h}{\partial t} + \boldsymbol{v}^h \cdot \nabla_{\boldsymbol{x}} \boldsymbol{v}^h + \nabla_{\boldsymbol{x}} p^h - \nu \Delta_{\boldsymbol{x}} \boldsymbol{v}^h - \boldsymbol{g} \tag{44.1}$$

$$r_C \left(\boldsymbol{v}^h\right) = \nabla_{\boldsymbol{x}} \cdot \boldsymbol{v}^h. \tag{44.2}$$

Proceeding this way, we obtain

$$B_{MS}^{ED}\left(\{\boldsymbol{w}^h, q^h\}, \{\boldsymbol{v}^h, p^h\}\right) - L^{ED}\left(\boldsymbol{w}^h\right) + B^{LD}\left(\boldsymbol{w}^h, \boldsymbol{v}^h\right) - L^{LD}\left(\boldsymbol{w}^h\right) = 0 \tag{45}$$

where

$$\begin{aligned} B_{MS}^{ED}\left(\{\boldsymbol{w}^h, q^h\}, \{\boldsymbol{v}^h, p^h\}\right) = & \left(\boldsymbol{w}^h(\boldsymbol{x}), \frac{\partial \boldsymbol{v}^h(\boldsymbol{x}, t)}{\partial t} + \boldsymbol{v}^h(\boldsymbol{x}, t) \cdot \nabla_{\boldsymbol{x}} \boldsymbol{v}^h(\boldsymbol{x}, t)\right)_{\Omega} \\ & - \left(\nabla_{\boldsymbol{x}} \cdot \boldsymbol{w}^h(\boldsymbol{x}), p^h(\boldsymbol{x})\right)_{\Omega} + \left(\nabla_{\boldsymbol{x}}^{\text{sym}} \boldsymbol{w}^h(\boldsymbol{x}), 2\nu \nabla_{\boldsymbol{x}}^{\text{sym}} \boldsymbol{v}^h(\boldsymbol{x}, t)\right)_{\Omega} \\ & + \left(q^h(\boldsymbol{x}), \nabla_{\boldsymbol{x}} \cdot \boldsymbol{v}^h(\boldsymbol{x}, t)\right)_{\Omega} - \left(\boldsymbol{w}^h(\boldsymbol{x}), \tau_M \boldsymbol{r}_M(\boldsymbol{u}^h, p^h) \cdot \nabla \boldsymbol{v}^h\right)_{\Omega} \\ & + \left(\nabla_{\boldsymbol{x}} \boldsymbol{w}^h(\boldsymbol{x}), (\boldsymbol{v}^h - \tau_M \boldsymbol{r}_M(\boldsymbol{v}^h, p^h)) \otimes \tau_M \boldsymbol{r}_M(\boldsymbol{v}^h, p^h)\right)_{\Omega} \\ & + \left(\nabla_{\boldsymbol{x}} \cdot \boldsymbol{w}^h(\boldsymbol{x}), \tau_C r_C(\boldsymbol{u}^h)\right)_{\Omega} + \left(\nabla_{\boldsymbol{x}} q^h, \tau_M \boldsymbol{r}_M(\boldsymbol{u}^h, p^h)\right)_{\Omega}. \end{aligned} \tag{46}$$

To compute the operator B^{LD} in Eq. (45) we need a discrete Lagrangian displacement that we define next. The discrete Lagrangian displacement lives in another discrete space $\mathcal{S}^h_{\overline{\boldsymbol{u}}^s} = \text{span}\{\overline{N}_B(\boldsymbol{X})\}_{B=1}^{n_{LD}}$, which is defined on the material description. Therefore, the discrete displacement can be written as

$$\overline{\boldsymbol{u}}^h(\boldsymbol{X}, t) = \sum_{B=1}^{n_{LD}} \overline{\boldsymbol{u}}_B(t) \overline{N}_B(\boldsymbol{X}). \tag{47}$$

Note that we have dropped the superscript s on the displacement, as all variables defined in material description refer to the solid, and thus, there is no possibility of confusion. Using Eq. (47) we can define a discrete mapping $\boldsymbol{\varphi}^h(\boldsymbol{X}, t)$ that will be used to push forward the solid mesh to physical space. Note that when pushed forward, the solid mesh will be located on top of the Eulerian mesh defined on Ω in an arbitrary, non-conforming fashion. All integrals on Ω_0^s (see the weak form (45)) will be computed on the mesh defined by the $\overline{N}_B$'s. Henceforth, this mesh will be called Lagrangian mesh because it is defined in the material description.[2]

Let us assume that the $\overline{N}_B$'s are defined from the tensor product of open knot vectors. Associated to those knot vectors, we define the set $\widehat{\mathcal{M}}^h$, which contains the Greville points in parametric coordinates. Greville points can be obtained by averaging knots, and a precise definition may be found in [75]. We will use the notation $\widehat{\mathcal{M}}^h = \{\widehat{\boldsymbol{\tau}}_i\}_{i=1}^{n_{LD}}$, where the $\widehat{\boldsymbol{\tau}}_i$'s are the Greville points in parametric space. For open knot vectors, $\#\widehat{\mathcal{M}}^h = \dim(\mathcal{S}^h_{\overline{\boldsymbol{u}}^s}) = n_{LD}$, where $\#\widehat{\mathcal{M}}^h$ denotes the cardinal of $\widehat{\mathcal{M}}^h$. Assuming that the initial solid geometry is mapped from the parametric space using a NURBS geometrical mapping $\boldsymbol{\phi}$, the Greville points in parametric space are mapped to the solid initial geometry as $\widetilde{\boldsymbol{\tau}}_i = \boldsymbol{\phi}(\widehat{\boldsymbol{\tau}}_i)$, defining the set $\widetilde{\mathcal{M}}^h = \{\widetilde{\boldsymbol{\tau}}_i\}_{i=1}^{n_{LD}}$. Then, at each time t, they are pushed forward to physical space using the mapping $\boldsymbol{\varphi}^h(\cdot, t)$. Let us call $\mathcal{M}^h = \{\boldsymbol{\tau}_i\}_{i=1}^{n_{LD}}$ the set of Greville points in physical space at time t, where

[2] Note that we always use NURBS elements and the nomenclature Lagrangian mesh should not induce the reader to think that we are using classical Lagrangian elements.

$\boldsymbol{\tau}_i = \boldsymbol{\varphi}^h(\widetilde{\boldsymbol{\tau}}_i, t)$. Using these ideas, we compute the Lagrangian displacement from the Eulerian velocity by collocating Eq. (20) at the Greville points as,

$$\frac{\partial \overline{\boldsymbol{u}}^h}{\partial t}(\widetilde{\boldsymbol{\tau}}_i, t) = \boldsymbol{v}^h(\boldsymbol{\varphi}^h(\widetilde{\boldsymbol{\tau}}_i, t), t) \quad \text{for all } \widetilde{\boldsymbol{\tau}}_i \in \widetilde{\mathcal{M}}^h. \tag{48}$$

Isogeometric collocation approaches have been recently introduced as a highly efficient alternative to Galerkin methods with higher-order convergence rates [70,71]. Within the realm of isogeometric collocation, there are several possibilities to select the collocation points [70]. Here, we use Greville points which have shown good accuracy and stability for all cases of practical importance. Note that expression (48) represents a system of dn_{LD} equations that we use to determine the dn_{LD} control variables of the discrete displacement $\overline{\boldsymbol{u}}^h$.

Remarks.

1. Eq. (44.1) contains second derivatives of $\boldsymbol{v}^h$. If the Eulerian mesh is C^0-continuous across element interfaces, a special procedure is needed for reconstructing second derivatives in order to avoid Dirac layers between elements [76]. However, we are able to evaluate second derivatives of $\boldsymbol{v}^h$ directly since all our Eulerian meshes are C^1-continuous, that is, we do not need any special procedure for reconstructing second derivatives.
2. Note that our space discretization is substantially different from those employed in [22,77]. In our formulation, the only variable that is defined on the Lagrangian space $\mathcal{S}^h_{\overline{\boldsymbol{u}}^s}$ is the displacement field $\overline{\boldsymbol{u}}^h$. For example, the term $\boldsymbol{v}^h(\boldsymbol{\varphi}^h(\boldsymbol{X}, t))$ of the operator B^{LD} is *not* projected onto $\mathcal{S}^h_{\overline{\boldsymbol{u}}^s}$, but it is directly evaluated as a function of the space $\mathcal{S}^h_v$. A similar argument applies, for example, to the term $\boldsymbol{w}^h(\boldsymbol{\varphi}^h(\boldsymbol{X}, t))$. In contrast, in the above-mentioned works [22,77], all the shape functions of the fluid Eulerian mesh need to be projected onto the solid mesh, and expressed as a linear combination of the $\overline{N}_B$'s. The computational cost of this projection is probably negligible using classical Lagrange finite elements, but represents a significant computational overhead for NURBS functions with degree $\mathsf{p} \geq 2$. In this case, the basis functions are not interpolatory and we need to solve a linear system of size n_{LD} to project each of the Eulerian basis functions onto the Lagrangian mesh. Our algorithm avoids this computational overhead altogether. For comparison purposes, we also implemented the isogeometric version of the approach proposed in [22,77], where all the basis functions of $\mathcal{S}^h_v$ are projected onto $\mathcal{S}^h_{\overline{\boldsymbol{u}}^s}$. For basis functions of degree $\mathsf{p} \geq 2$, our algorithm was significantly faster and, at the very least, as accurate as that based on the projection of Eulerian basis functions. Earlier works also report on the fact that, according to numerical evidence, the Lagrangian mesh needs to be at least twice as fine as the Eulerian mesh. As suggested in [77], this could be related to the projection of the fluid shape functions onto the space $\mathcal{S}^h_{\overline{\boldsymbol{u}}^s}$. In some situations, the projection on $\mathcal{S}^h_{\overline{\boldsymbol{u}}^s}$ of Eulerian shape functions with significant support on the solid is zero if the Lagrangian mesh is not fine enough. As we will show in our numerical examples, our method also circumvents this problem, and we have been able to perform very accurate computations on Lagrangian meshes which have the same element size as the Eulerian mesh. This is also related to the fact that the order of the Eulerian meshes that we utilize in our simulations is $\mathsf{p} \geq 2$ and, as a result, the support of the basis functions is $\mathsf{p}+1 \geq 3$ elements in each direction. Since the support of the basis functions is larger, the above-described situation will be probably avoided.
3. From an implementation point of view, the semi-discrete weak form (45) is computed as follows: We first loop over the elements of the Eulerian mesh defined on Ω to compute the terms B^{ED}_{MS} and L^{ED}. Then, we loop over the elements of the Lagrangian mesh defined on Ω^s_0 to subtract the fluid contributions and add the terms of the solid mechanics equations. One of the crucial advantages of this implementation is that when we loop over the elements on Ω we do not need to know where the solid is located with respect to the Eulerian mesh to compute the integrals.
4. Following [77], when we loop over the Lagrangian mesh to subtract the contributions of the fluid mechanics equations computing B^{LD} the stabilization terms are not subtracted. Note also that the residuals from which we define the velocity and pressure fine scales (44) are those of the fluid mechanics equations, which correspond to the right-hand sides of Eq. (44). The residual r_C will approach zero on the whole computational domain, including the solid domain as the whole system is incompressible. However, the residual $\boldsymbol{r}_M$ will approach zero within the fluid domain, but not within the solid domain. Therefore, we do not have a residual-based stabilization in the classical sense of this term. We do note, nevertheless, that our stabilization is consistent in the entire fluid–solid system because τ_M tends to zero as the mesh is refined. This solution is less than satisfactory for us, but computing the residuals in a fully consistent way would produce a significant computational overhead, and would require

an algorithmical tracking on the solid position when we loop over the Eulerian mesh. This can be incorporated into our algorithm, but one of its most appealing features would be lost. Another way to make the stabilization completely residual based would be not to construct the stabilization terms of the fluid mechanics equations in the elements covered by the structure. This would introduce inefficiencies in the algorithm and would require to construct suitable stabilization terms in the solid to deal with the incompressibility constraint. For these reasons, we did not consider this option, either. In practical computations, the behavior of our algorithm has been very satisfactory, but we acknowledge that with this implementation it may not be possible to achieve higher-order convergence rates within the solid domain. We believe this point deserves more investigation, and we hope to address it in the future.

3.2. Time discretization

At this stage our formulation remains continuous in time. Here, we propose a monolithic and fully-implicit algorithm based on the generalized-α method. The generalized-α algorithm was first proposed by Chung and Hulbert [62] for the structural mechanics equations, and later extended by Jansen et al. [63] to first-order systems. To illustrate our method, let us divide the time interval of interest $[0, T]$ into a sequence of subintervals (t_n, t_{n+1}) with fixed time-step size $\Delta t = t_{n+1} - t_n$. To describe our algorithm, we define the following residual vectors

$$\boldsymbol{R}^M = \left\{ R^M_{A,i} \right\}; \qquad \boldsymbol{R}^C = \left\{ R^C_A \right\}; \qquad \boldsymbol{R}^P = \left\{ R^P_{B,i} \right\}. \tag{49}$$

Here, $A \in \{1, \ldots, n_{ED}\}$ is a control-variable index, B is an index that takes values from 1 to n_{LD}, and i is a dimension index which runs from 1 to d. The components of the residual vectors are given by

$$R^M_{A,i} = B^{ED}_{MS}\left(\{N_A \boldsymbol{e}_i, 0\}, \{\boldsymbol{v}^h, p^h\}\right) - L^{ED}\left(N_A \boldsymbol{e}_i\right) + B^{LD}\left(N_A \boldsymbol{e}_i, \boldsymbol{v}^h\right) - L^{LD}\left(N_A \boldsymbol{e}_i\right) \tag{50}$$

$$R^C_A = B^{ED}_{MS}\left(\{0, N_A\}, \{\boldsymbol{v}^h, p^h\}\right) \tag{51}$$

$$R^P_{B,i} = \boldsymbol{e}_i \cdot \left[\frac{\partial \overline{\boldsymbol{u}}^h}{\partial t}(\widetilde{\boldsymbol{\tau}}_B, t) - \boldsymbol{v}^h\left(\boldsymbol{\varphi}^h(\widetilde{\boldsymbol{\tau}}_B, t), t\right) \right]. \tag{52}$$

Let us call $\boldsymbol{V}_n, \boldsymbol{P}_n$ and $\boldsymbol{A}_n$ the time-discrete approximation of the global vector of control variables of $\boldsymbol{v}^h(\cdot, t_n)$, $p^h(\cdot, t_n)$, and $\frac{\partial \boldsymbol{v}^h}{\partial t}(\cdot, t_n)$, respectively. Similarly, $\overline{\boldsymbol{U}}_n$ and $\overline{\boldsymbol{V}}_n$ are the time-discrete approximation of the global vector of control variables of $\overline{\boldsymbol{u}}^h(\cdot, t_n)$ and $\frac{\partial \overline{\boldsymbol{u}}^h}{\partial t}(\cdot, t_n)$. Let us also define $\overline{\boldsymbol{U}}^G_n$ as the vector that collects the time-discrete approximation to the solid displacements at the Greville points, that is, $\overline{\boldsymbol{U}}^G_n \approx \{\overline{u}^h_i(\widetilde{\boldsymbol{\tau}}_A, t_n)\}^{n_{LD}}_{A=1}$ for all $i = 1, \ldots, d$. Analogously, we define $\overline{\boldsymbol{V}}^G_n \approx \{\frac{\partial \overline{u}^h_i}{\partial t}(\widetilde{\boldsymbol{\tau}}_A, t_n)\}^{n_{LD}}_{A=1}$ for all $i = 1, \ldots, d$. Using this notation, our time-integration algorithm may be defined as follows: given $\boldsymbol{V}_n, \boldsymbol{A}_n, \overline{\boldsymbol{U}}_n$, and $\overline{\boldsymbol{V}}_n$, find $\boldsymbol{V}_{n+1}, \boldsymbol{A}_{n+1}, \boldsymbol{V}_{n+\alpha_f}, \boldsymbol{A}_{n+\alpha_m}, \boldsymbol{P}_{n+1}, \overline{\boldsymbol{U}}_{n+1}$, $\overline{\boldsymbol{V}}_{n+1}, \overline{\boldsymbol{U}}_{n+\alpha_f}$, and $\overline{\boldsymbol{V}}_{n+\alpha_m}$ such that

$$\boldsymbol{R}^M(\boldsymbol{V}_{n+\alpha_f}, \boldsymbol{A}_{n+\alpha_m}, \boldsymbol{P}_{n+1}, \overline{\boldsymbol{U}}^G_{n+\alpha_f}) = 0 \tag{53}$$

$$\boldsymbol{R}^C(\boldsymbol{V}_{n+\alpha_f}, \boldsymbol{A}_{n+\alpha_m}, \boldsymbol{P}_{n+1}) = 0 \tag{54}$$

$$\boldsymbol{R}^P(\overline{\boldsymbol{U}}^G_{n+\alpha_f}, \overline{\boldsymbol{V}}^G_{n+\alpha_m}, \boldsymbol{V}_{n+\alpha_f}) = 0 \tag{55}$$

$$\boldsymbol{V}_{n+\alpha_f} = \boldsymbol{V}_n + \alpha_f(\boldsymbol{V}_{n+1} - \boldsymbol{V}_n) \tag{56}$$

$$\boldsymbol{A}_{n+\alpha_m} = \boldsymbol{A}_n + \alpha_m(\boldsymbol{A}_{n+1} - \boldsymbol{A}_n) \tag{57}$$

$$\overline{\boldsymbol{U}}^G_{n+\alpha_f} = \overline{\boldsymbol{U}}^G_n + \alpha_f(\overline{\boldsymbol{U}}^G_{n+1} - \overline{\boldsymbol{U}}^G_n) \tag{58}$$

$$\overline{\boldsymbol{V}}^G_{n+\alpha_m} = \overline{\boldsymbol{V}}^G_n + \alpha_m(\overline{\boldsymbol{V}}^G_{n+1} - \overline{\boldsymbol{V}}^G_n) \tag{59}$$

$$\boldsymbol{V}_{n+1} = \boldsymbol{V}_n + \Delta t((1-\gamma)\boldsymbol{A}_n + \gamma \boldsymbol{A}_{n+1}) \tag{60}$$

$$\overline{\boldsymbol{U}}^G_{n+1} = \overline{\boldsymbol{U}}^G_n + \Delta t((1-\gamma)\overline{\boldsymbol{V}}^G_n + \gamma \overline{\boldsymbol{V}}^G_{n+1}). \tag{61}$$

Note that although $\boldsymbol{V}_{n+1}$ and $\boldsymbol{A}_{n+1}$ are treated separately, they are not independent as Eq. (60) must hold true. The same argument applies to $\overline{\boldsymbol{U}}^G_{n+1}$ and $\overline{\boldsymbol{V}}^G_{n+1}$, which are related through Eq. (61). In Eqs. (53)–(61), α_m, α_f and γ are real-valued parameters that control the accuracy and stability of the algorithm. Jansen et al. [63] showed that second-order accuracy can be attained by taking

$$\gamma = \frac{1}{2} + \alpha_m - \alpha_f, \tag{62}$$

while unconditional stability (for a linear problem) requires

$$\alpha_m \geq \alpha_f \geq \frac{1}{2}. \tag{63}$$

A particular feature of generalized-α is that the method parameters can be expressed in terms of ϱ_∞ (the spectral radius of the amplification matrix as $\Delta t \to \infty$) in such a way that conditions (62)–(63) are automatically satisfied. This can be accomplished by taking

$$\alpha_m = \frac{1}{2}\left(\frac{3-\varrho_\infty}{1+\varrho_\infty}\right) \tag{64}$$

$$\alpha_f = \gamma = \frac{1}{1+\varrho_\infty} \tag{65}$$

with $\varrho_\infty \in [0, 1]$. By using Eqs. (64)–(65), all the eigenvalues of the amplification matrix take on the value $-\varrho_\infty$ when $\Delta t \to \infty$, which suggests that ϱ_∞ can be used to control high-frequency dissipation with the guarantee that second-order accuracy and unconditional stability for a linear problem are granted. In our computations, we have taken $\varrho_\infty = 1/2$, which represents an adequate balance between accuracy and robustness in our simulations.

3.3. Implementation

Eqs. (53)–(55) constitute a nonlinear system. To solve this nonlinear system we use Newton–Raphson's algorithm which results in a two-stage predictor multicorrector method defined as follows:

Predictor stage: Set

$$\boldsymbol{V}_{n+1,(0)} = \boldsymbol{V}_n \tag{66}$$

$$\boldsymbol{A}_{n+1,(0)} = \frac{(\gamma-1)}{\gamma}\boldsymbol{A}_n \tag{67}$$

$$\boldsymbol{P}_{n+1,(0)} = \boldsymbol{P}_n \tag{68}$$

$$\overline{\boldsymbol{U}}_{n+1,(0)} = \overline{\boldsymbol{U}}_n \tag{69}$$

$$\overline{\boldsymbol{V}}_{n+1,(0)} = \frac{(\gamma-1)}{\gamma}\overline{\boldsymbol{V}}_n \tag{70}$$

where the subscript 0 on the left-hand side quantities is the Newton–Raphson iteration index. Note that the predictions are based on Eqs. (66), (68) and (69), while (67) and (70) are only a consequence of Eqs. (60) and (61).

Multicorrector stage: Repeat the following steps for $i = 0, 1, 2, \ldots, i_{\max}$, or until convergence is achieved:

1. Evaluate the global unknowns at intermediate time levels

$$\boldsymbol{V}_{n+\alpha_f,(i)} = \boldsymbol{V}_n + \alpha_f(\boldsymbol{V}_{n+1,(i)} - \boldsymbol{V}_n) \tag{71}$$

$$\boldsymbol{A}_{n+\alpha_m,(i)} = \boldsymbol{A}_n + \alpha_m(\boldsymbol{A}_{n+1,(i)} - \boldsymbol{A}_n) \tag{72}$$

$$\boldsymbol{P}_{n+1,(i)} = \boldsymbol{P}_{n+1,(i)} \tag{73}$$

$$\overline{\boldsymbol{U}}_{n+\alpha_f,(i)} = \overline{\boldsymbol{U}}_n + \alpha_f(\overline{\boldsymbol{U}}_{n+1,(i)} - \overline{\boldsymbol{U}}_n) \tag{74}$$

$$\overline{\boldsymbol{V}}_{n+\alpha_m,(i)} = \overline{\boldsymbol{V}}_n + \alpha_m(\overline{\boldsymbol{V}}_{n+1,(i)} - \overline{\boldsymbol{V}}_n). \tag{75}$$

2. Use the intermediate time levels of the ith Newton iteration to compute the tangent matrix $\boldsymbol{K}_{,(i)}$ and the residual vector $\boldsymbol{R}_{,(i)}$, where

$$\boldsymbol{K}_{,(i)} = \begin{pmatrix} \boldsymbol{K}_{11,(i)} & \boldsymbol{K}_{12,(i)} & \boldsymbol{K}_{13,(i)} \\ \boldsymbol{K}_{21,(i)} & \boldsymbol{K}_{22,(i)} & \boldsymbol{K}_{23,(i)} \\ \boldsymbol{K}_{31,(i)} & \boldsymbol{K}_{32,(i)} & \boldsymbol{K}_{33,(i)} \end{pmatrix}; \qquad \boldsymbol{R}_{,(i)} = \begin{Bmatrix} \boldsymbol{R}^M_{,(i)} \\ \boldsymbol{R}^C_{,(i)} \\ \boldsymbol{R}^P_{,(i)} \end{Bmatrix}. \tag{76}$$

Note that the residual $\boldsymbol{R}_{,(i)}$ may be linearized with respect to different variables, leading to distinct tangent matrices. Here, we linearize with respect to $\boldsymbol{A}_{n+1}$, $\boldsymbol{P}_{n+1}$ and $\overline{\boldsymbol{V}}^G_{n+1}$. Once we have decided to linearize with respect to $\boldsymbol{A}_{n+1}$ and $\overline{\boldsymbol{V}}^G_{n+1}$, the variables $\boldsymbol{V}_{n+1}$ and $\overline{\boldsymbol{U}}^G_{n+1}$ as well as all the α-levels become linearly dependent upon $\boldsymbol{A}_{n+1}$ and $\overline{\boldsymbol{V}}^G_{n+1}$ through Eqs. (56)–(61). Using this choice, the tangent matrix is given by

$$\boldsymbol{K}_{11,(i)} = \frac{\partial \boldsymbol{R}^M_{,(i)}}{\partial \boldsymbol{A}_{n+1,(i)}}; \qquad \boldsymbol{K}_{12,(i)} = \frac{\partial \boldsymbol{R}^M_{,(i)}}{\partial \boldsymbol{P}_{n+1,(i)}}; \qquad \boldsymbol{K}_{13,(i)} = \frac{\partial \boldsymbol{R}^M_{,(i)}}{\partial \overline{\boldsymbol{V}}^G_{n+1,(i)}}; \tag{77}$$

$$\boldsymbol{K}_{21,(i)} = \frac{\partial \boldsymbol{R}^C_{,(i)}}{\partial \boldsymbol{A}_{n+1,(i)}}; \qquad \boldsymbol{K}_{22,(i)} = \frac{\partial \boldsymbol{R}^C_{,(i)}}{\partial \boldsymbol{P}_{n+1,(i)}}; \qquad \boldsymbol{K}_{23,(i)} = \frac{\partial \boldsymbol{R}^C_{,(i)}}{\partial \overline{\boldsymbol{V}}^G_{n+1,(i)}}; \tag{78}$$

$$\boldsymbol{K}_{31,(i)} = \frac{\partial \boldsymbol{R}^P_{,(i)}}{\partial \boldsymbol{A}_{n+1,(i)}}; \qquad \boldsymbol{K}_{32,(i)} = \frac{\partial \boldsymbol{R}^P_{,(i)}}{\partial \boldsymbol{P}_{n+1,(i)}}; \qquad \boldsymbol{K}_{33,(i)} = \frac{\partial \boldsymbol{R}^P_{,(i)}}{\partial \overline{\boldsymbol{V}}^G_{n+1,(i)}}. \tag{79}$$

We note that $\boldsymbol{K}_{23}$ and $\boldsymbol{K}_{32}$ vanish. In our implementation, we neglect the contributions $\boldsymbol{K}_{13}$ and $\boldsymbol{K}_{31}$ to the tangent matrix. Under these assumptions, and considering that $\boldsymbol{K}_{23}$ and $\boldsymbol{K}_{32}$ are zero, the global linear system may be decoupled as

$$\begin{pmatrix} \boldsymbol{K}_{11,(i)} & \boldsymbol{K}_{12,(i)} \\ \boldsymbol{K}_{21,(i)} & \boldsymbol{K}_{22,(i)} \end{pmatrix} \begin{Bmatrix} \Delta \boldsymbol{A}_{n+1,(i)} \\ \Delta \boldsymbol{P}_{n+1,(i)} \end{Bmatrix} = - \begin{Bmatrix} \boldsymbol{R}^M_{,(i)} \\ \boldsymbol{R}^C_{,(i)} \end{Bmatrix} \tag{80}$$

$$\boldsymbol{K}_{33,(i)} \Delta \overline{\boldsymbol{V}}^G_{n+1,(i)} = -\boldsymbol{R}^P_{,(i)} \tag{81}$$

with the additional advantage that the unknowns of each Greville point in the global vector $\Delta \overline{\boldsymbol{V}}^G_{n+1,(i)}$ are decoupled in such a way that (81) is solved as n_{LD} independent systems of equations of size $d \times d$. We remark that the matrix $\boldsymbol{K}_{33,(i)}$ has this special shape because we are linearizing with respect to the displacements at Greville points $\overline{\boldsymbol{U}}^G_{n+1}$. If we were to linearize with respect to their associated control variables $\overline{\boldsymbol{U}}_{n+1}$, the dn_{LD} equations would be coupled, leading to a more intensive computation.

The submatrices $\boldsymbol{K}_{lm}$ for $l, m \in \{1, 2, 3\}$ may be computed using the chain rule. For example,

$$\begin{aligned} \boldsymbol{K}_{11,(i)} &= \frac{\partial \boldsymbol{R}^M_{,(i)}}{\partial \boldsymbol{A}_{n+1,(i)}} \\ &= \frac{\partial \boldsymbol{R}^M_{,(i)}}{\partial \boldsymbol{V}_{n+\alpha_f,(i)}} \frac{\partial \boldsymbol{V}_{n+\alpha_f,(i)}}{\partial \boldsymbol{V}_{n+1,(i)}} \frac{\partial \boldsymbol{V}_{n+1,(i)}}{\partial \boldsymbol{A}_{n+1,(i)}} + \frac{\partial \boldsymbol{R}^M_{,(i)}}{\partial \boldsymbol{A}_{n+\alpha_m,(i)}} \frac{\partial \boldsymbol{A}_{n+\alpha_m,(i)}}{\partial \boldsymbol{A}_{n+1,(i)}} \\ &= \alpha_f \gamma \Delta t \frac{\partial \boldsymbol{R}^M_{,(i)}}{\partial \boldsymbol{V}_{n+\alpha_f,(i)}} + \alpha_m \frac{\partial \boldsymbol{R}^M_{,(i)}}{\partial \boldsymbol{A}_{n+\alpha_m,(i)}}. \end{aligned} \tag{82}$$

Finally, we note that the global linear system (80) has the same size as that of a fully-implicit incompressible Navier–Stokes formulation and is solved up to a given tolerance using the GMRES method [78] with incomplete LU factorization [79–81].

3. Update the Newton–Raphson iterates as follows

$$\boldsymbol{V}_{n+1,(i+1)} = \boldsymbol{V}_{n+1,(i)} + \gamma \Delta t \Delta \boldsymbol{A}_{n+1,(i)} \tag{83}$$

$$\boldsymbol{A}_{n+1,(i+1)} = \boldsymbol{A}_{n+1,(i)} + \Delta \boldsymbol{A}_{n+1,(i)} \tag{84}$$

$$\boldsymbol{P}_{n+1,(i+1)} = \boldsymbol{P}_{n+1,(i)} + \Delta \boldsymbol{P}_{n+1,(i)} \tag{85}$$

$$\overline{\boldsymbol{U}}^{G}_{n+1,(i+1)} = \overline{\boldsymbol{U}}^{G}_{n+1,(i)} + \gamma\,\Delta t\,\Delta\overline{\boldsymbol{V}}^{G}_{n+1,(i)} \tag{86}$$

$$\overline{\boldsymbol{V}}^{G}_{n+1,(i+1)} = \overline{\boldsymbol{V}}^{G}_{n+1,(i)} + \Delta\overline{\boldsymbol{V}}^{G}_{n+1,(i)}. \tag{87}$$

Note that the updates of $\boldsymbol{A}_{n+1}$, $\boldsymbol{P}_{n+1}$ and $\overline{\boldsymbol{V}}^{G}_{n+1}$ are straightforward while $\boldsymbol{V}_{n+1}$ and $\overline{\boldsymbol{U}}^{G}_{n+1}$ are updated consistently with Eqs. (60)–(61) that define the generalized-α method.

4. From the ith Newton iterate of the time-discrete Greville values of the solid displacement $\overline{\boldsymbol{U}}^{G}_{n+1,(i)}$, compute their corresponding control variables $\overline{\boldsymbol{U}}_{n+1,(i)}$. This may be done independently for each of the spatial directions. Therefore, let us define $\overline{\boldsymbol{\mathcal{U}}}^{G}_{n+1,(i)}$ as the entries of $\overline{\boldsymbol{U}}^{G}_{n+1,(i)}$ that correspond to the first spatial direction. Similarly, we define $\overline{\boldsymbol{\mathcal{U}}}_{n+1,(i)}$ as the restriction of $\overline{\boldsymbol{U}}_{n+1,(i)}$ to the first spatial dimension. Then, we can compute $\overline{\boldsymbol{\mathcal{U}}}_{n+1,(i)}$ by solving the linear system

$$\boldsymbol{M}\overline{\boldsymbol{\mathcal{U}}}_{n+1,(i)} = \overline{\boldsymbol{\mathcal{U}}}^{G}_{n+1,(i)} \tag{88}$$

where the entries of the matrix $\boldsymbol{M}$ are given by

$$\boldsymbol{M} = \{M_{AB}\}; \qquad M_{AB} = \overline{N}_B(\widetilde{\boldsymbol{\tau}}_A); \quad A, B \in \{1, \ldots, n_{LD}\}. \tag{89}$$

Note that the matrix $\boldsymbol{M}$ depends neither on time nor on the spatial direction. Thus, $\boldsymbol{M}$ can be pre-computed and pre-factorized at the beginning of the simulation. Whenever we need to solve system (88), we only perform the corresponding backward and forward substitutions. The process defined by Eqs. (88)–(89) is repeated for the remaining spatial directions. This completes one nonlinear iteration.

Remark. We acknowledge that our strategy of using a non-consistent tangent matrix (see Step 2 of the Multicorrector stage) may limit the size of our time steps. However, computing the consistent tangent matrix produces a significant computational overhead, so we chose not to do so. Other authors have reported on the use of matrix-free Newton–Krylov methods [22]. In our implementation, utilizing the tangent matrix (80)–(81) vastly outperformed the use of matrix-free techniques. Using matrix-free methods, the computational cost of a time step was higher and we could not take time steps as large as those employed with the tangent matrix (80)–(81). In the future, we will investigate in more detail the possibility of using the consistent tangent matrix.

4. Numerical examples

In this section we present four numerical examples computed using the discretization introduced in Sections 3.1 and 3.2. The code used to perform these simulations has been developed on top of the PetIGA framework [82–84], which adds NURBS discretization capabilities and integration of forms to the scientific library PETSc [85,86].

4.1. Free falling cylinder

An object falling in a fluid will try to adjust its velocity to a terminal or settling velocity that we call v_T. The terminal velocity of a falling object is reached when the sum of the drag force (F_D), which increases with velocity, and buoyancy (F_b) equals the downward force of gravity (F_g) acting on the object. Since the net force on the object is then zero, the object moves with constant velocity. That is, to obtain the exact solution for the settling velocity of a rigid cylinder with infinite length and radius a which is released to fall in a channel of width $2L$, we impose

$$F_D + F_b = F_g \tag{90}$$

where

$$F_b = g\rho^f \pi a^2 \tag{91}$$

$$F_g = g\rho^s \pi a^2. \tag{92}$$

Here, g is the Euclidean norm of $\boldsymbol{g}$. We assume that for this problem the relevant Reynolds number is based on the terminal velocity and the cylinder diameter, that is,

$$R_e = \frac{2\rho^f v_T a}{\mu}. \tag{93}$$

According to [87], under the assumption of creeping flow ($R_e \ll 1$), and undeformable solid, F_D can be expressed as

$$F_D = \frac{4\pi\mu v_T}{\ln\left(\frac{L}{a}\right) - 0.9157 + 1.7244\left(\frac{a}{L}\right)^2 - 1.7302\left(\frac{a}{L}\right)^4}, \tag{94}$$

which using Eqs. (90)–(92) leads to the terminal velocity

$$v_T = \frac{\left(\rho^s - \rho^f\right) g a^2}{4\mu}\left[\ln\left(\frac{L}{a}\right) - 0.9157 + 1.7244\left(\frac{a}{L}\right)^2 - 1.7302\left(\frac{a}{L}\right)^4\right]. \tag{95}$$

In this numerical example, we will validate our numerical results against the exact solution (95). The infinite length of the cylinder allows us to neglect the dynamics in the direction of the axis of the cylinder and therefore we set up a two-dimensional domain, which has to be long enough in the direction of gravity such that the cylinder can reach its terminal velocity. The solid has to be sufficiently stiff in order to complete the whole simulation without experiencing perceptible deformations. We have used a Neo-Hookean model with dilatational penalty in 2D, whose stored energy function can be obtained by setting $d = 2$ in Eqs. (16)–(18). Taking the derivative of the stored energy function we get the second Piola–Kirchhoff stress tensor

$$\boldsymbol{S} = \mu^s J^{-2/d}\left(\boldsymbol{I} - \frac{1}{d}\mathrm{tr}(\boldsymbol{C})\boldsymbol{C}^{-1}\right) + \frac{1}{2}\kappa^s\left(J^2 - 1\right)\boldsymbol{C}^{-1} \tag{96}$$

where $d = 2$. Note that following [21], for this example, and also for the remaining examples, we have included also ψ^{dil} into the stored energy functional to derive $\boldsymbol{S}$. Consistently with what was reported in [21], we found that disregarding the dilatational part of the stored energy function produced non-negligible errors in the incompressibility constraint. Thus, following [21] we used relatively small values of κ^s for our computations (on the order of magnitude of $\mu^s/10$). Within this range, we did not find the numerical solution to be sensitive to the value of κ^s. The remaining physical parameters are taken as follows: gravity acceleration $g = 981$ cm/s^2, dynamic viscosity $\mu = 5$ dyn/cm^2, fluid density $\rho^f = 1$ g/cm^3, solid density $\rho^s = 1.25$ g/cm^3 and shear modulus $\mu^s = 33\,550$ g/(cms^2). The dimensions of the computational domain are $2L \times 3L$ with $L = 2$ cm and the radius of the cylinder is $a = 0.25$ cm. Introducing these parameters in Eq. (95), we obtain $v_{\mathrm{T}} = 0.912$ cm/s and the Reynolds number is $R_e = 0.0912 \ll 1$.

The geometry of the problem, boundary conditions and Lagrangian mesh are represented in Fig. 2. NURBS shape functions, unlike Lagrange shape functions, are able to represent the axial section of a cylinder exactly when NURBS of second order or higher are used. Note that one face of each of the elements adjacent to the cylinder axis has been degenerated by placing multiple control points at the same location, and thus many parameter values map to the same point in physical space. Such a mapping is not invertible, but it is still analysis-suitable since the Gauss quadrature points never fall on the singularity itself.

Both the Lagrangian mesh and the Eulerian mesh are comprised of quadratic NURBS elements. The shape functions of the Eulerian mesh are at least $\mathcal{C}^1$-continuous everywhere. The shape functions of the Lagrangian mesh are $\mathcal{C}^0$-continuous (in the circumferential direction) along two diametral lines, and at least $\mathcal{C}^1$-continuous everywhere else. All the simulations are performed with the same Lagrangian mesh which has 11 elements in the radial direction and 48 elements in the circumferential direction. The time step used in the simulations is $\Delta t = 10^{-3}$ s. We have computed the average solid velocity in the direction of gravity for simulations with different Eulerian meshes. The points chosen to calculate that average are the images of the Greville points. As shown in Fig. 3, the more we refine the Eulerian mesh, the closer we get to the analytical solution. The relative difference between the analytical and numerical terminal velocity is 5.7% with 100×150 elements, 2.1% with 150×225 elements, and 0.5% with 200×300 elements.

Fig. 4 shows the vortices and the recirculation that appear close to the free falling cylinder. In Fig. 5, we plot the pressure subtracting its hydrostatic component along with the Lagrangian mesh and the finest Eulerian mesh used in our computations. It is noticeable that the elements of the Lagrangian mesh are even bigger than the elements of the Eulerian mesh. Despite this fact, the simulation yields very accurate results as it can be seen in Fig. 3. Based on previously reported numerical results, the possibility of using a Lagrangian mesh coarser than the Eulerian mesh seems to be a unique feature of NURBS functions with $\mathsf{p} \geq 2$ that is not attained with classical Lagrange elements.

In Fig. 6, we consider several fluid viscosities while the rest of parameters remain fixed. We then compare the numerical result with the analytical solution for each case. In Fig. 7, we repeat the same procedure, but this time we

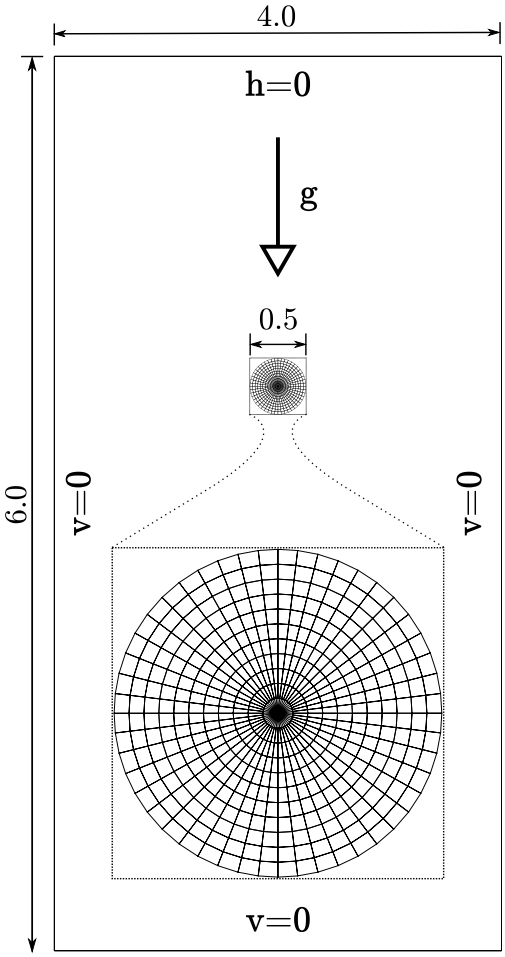


Fig. 2. Geometry, boundary conditions, and Lagrangian mesh for the free falling cylinder example. Note that we impose non-slip boundary conditions on the lateral boundaries as this is the case for which the exact solution applies. The inset shows a zoom in of the Lagrangian mesh. The lengths are in centimeters.

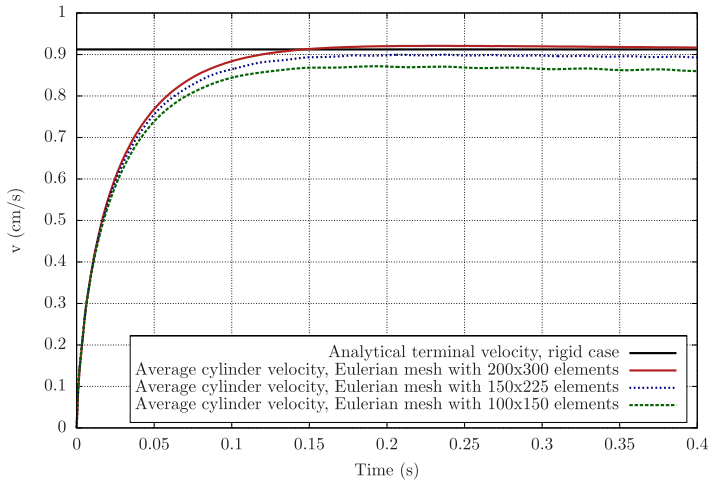


Fig. 3. Exact terminal velocity (black line) and time evolution of the average cylinder velocity obtained from the computations (red, blue and green lines). All computations were performed using a Lagrangian mesh composed of 11 × 48 elements (knot spans). The Eulerian mesh was refined to show convergence of the numerical solution to the exact solution. In all cases, we used quadratic basis functions. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

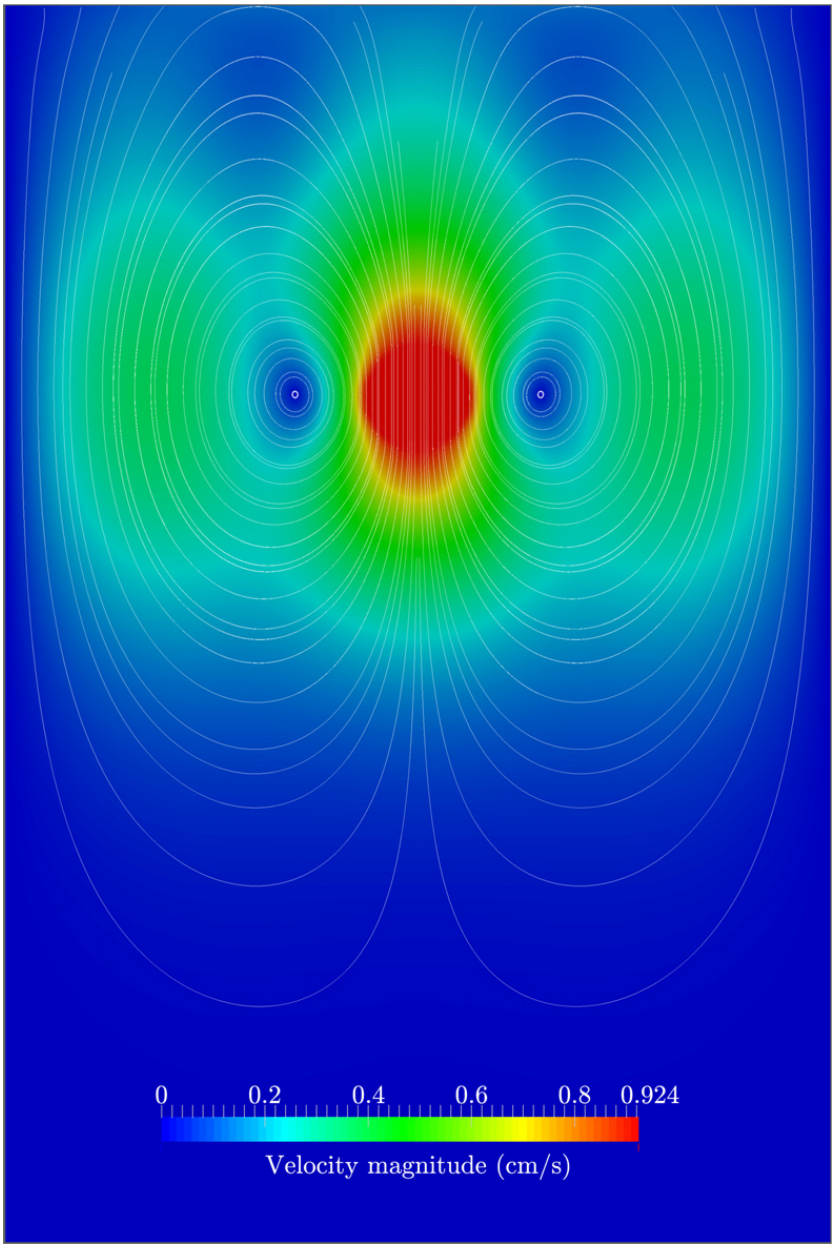


Fig. 4. (Color online) Velocity magnitude along with the streamlines at time $t = 0.35$ s for the free-falling cylinder example. The plot clearly shows two large vortices close to the cylinder, which is the expected laminar-flow solution. The Eulerian mesh is composed of 200×300 elements and the Lagrangian mesh has 11×48 elements. In all cases, we used quadratic basis functions. At the scale of the plot, the solution looks symmetric as expected.

vary the solid density while keeping the other parameters fixed. In both cases, the difference between the numerical result and the analytical solution increases as the Reynolds number increases. This comes as no surprise, since the analytical solution is deduced under the assumption of creeping flow and as a result it looses its accuracy for predicting the correct terminal velocity when the Reynolds number gets close to one or is even higher. This is precisely what Figs. 6 and 7 show.

Finally, for comparison purposes, we also computed this example by using an alternative algorithm that projects the shape functions of the Eulerian mesh onto the Lagrangian mesh using a collocation-like process for each shape function (see Remark 2 in Section 3.1). The idea is to project the Eulerian shape functions using a procedure similar to that used for the displacement field. Note that the projection of a shape function, though, is a linear process, unlike the projection of the displacement, which is nonlinear. We have also tried projecting the Eulerian velocity and its time derivative and using these projected quantities when looping over the elements of the Lagrangian mesh. For that, we

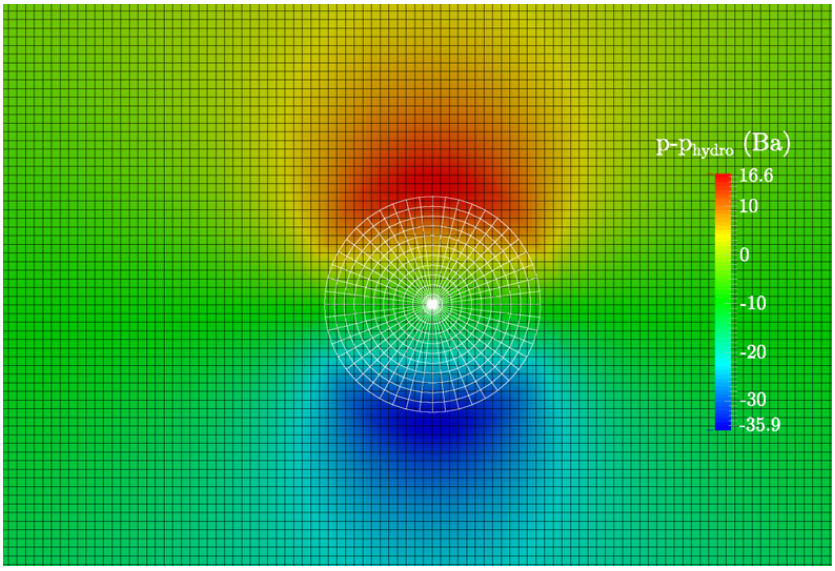


Fig. 5. (Color online) Contour plot of the pressure (subtracting the hydrostatic component) along with the spatial discretizations. The Eulerian mesh is composed of 200 × 300 elements and the Lagrangian mesh has 11 × 48 elements. In all cases, we used quadratic basis functions. Note that the elements of the Lagrangian mesh (white lines) are even bigger than the elements of the Eulerian mesh (black lines), and, yet, the solution is very accurate as shown in Fig. 3.

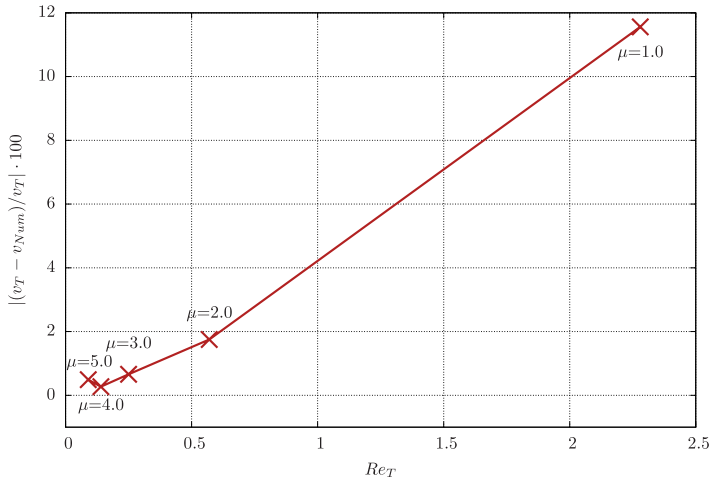


Fig. 6. Relative difference between the analytical and numerical terminal velocities for several values of the fluid viscosity. The analytical solution is only acceptable when $R_e \ll 1$ since it is a solution for creeping flow. When $R_e \ll 1$, the analytical and numerical solutions show very good agreement. As the Reynolds number gets close to one or higher, the difference between the two solutions increases as it must happen. The Eulerian mesh has 200 × 300 elements and the Lagrangian mesh has 11 × 48 elements. We used quadratic basis functions.

have made use of Eq. (9). The variation of the results when using these projected quantities is very subtle (data not shown) and the algorithm becomes slower.

4.2. *Free falling sphere*

The second example is a 3D computation of a sphere of radius a falling in a fluid. The terminal velocity of the simulation is going to be compared with an analytical solution in order to validate the method in a three-dimensional setup. As it has been mentioned in the previous section, we impose Eq. (90). Although this time the drag force, buoyancy and the downward force of gravity acting on the object are given by

$$F_D = \frac{1}{2}\pi a^2 C_D \rho^f v_T^2 \tag{97}$$

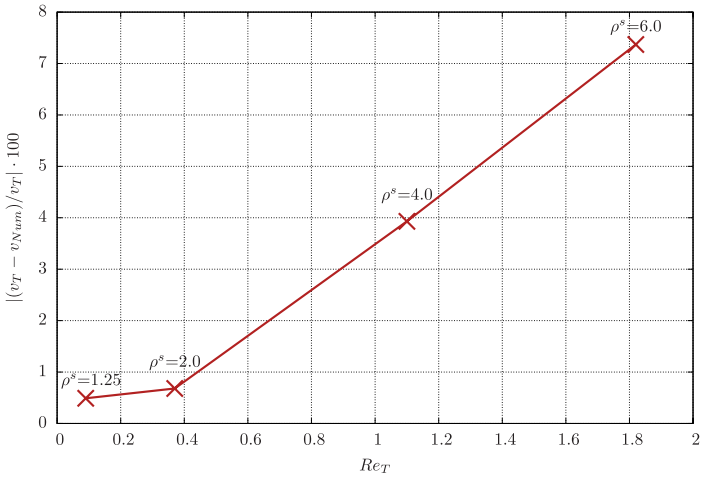


Fig. 7. Relative error of the numerical terminal velocity for several values of the solid density. The exact solution is only acceptable when $R_e \ll 1$ since it is a solution for creeping flow. When $R_e \ll 1$, the exact and numerical solutions show good agreement. As the Reynolds number gets close to one or higher, the difference between the two solutions increases because the theoretical solution becomes inaccurate. The Eulerian mesh is composed of 200 × 300 elements and the Lagrangian mesh has 11 × 48 elements. We used quadratic basis functions.

$$F_b = \frac{4}{3}\pi a^3 g \rho^f \tag{98}$$

$$F_g = \frac{4}{3}\pi a^3 g \rho^s. \tag{99}$$

If we knew the value of the drag coefficient C_D, we could calculate the terminal velocity of the free falling sphere. Theoretical solutions under the assumption of creeping flow ($R_e \ll 1$) show that the drag coefficient of a sphere falling in an unbounded domain takes the value

$$C_D = \frac{24}{R_e} \tag{100}$$

where R_e is the same as in Eq. (93), but this time a refers to the radius of the sphere. This expression for the drag coefficient yields the terminal velocity

$$v_{Stokes} = \frac{2g}{9\mu} a^2 (\rho^s - \rho^f). \tag{101}$$

We have renamed v_T to v_{Stokes} not only because this solution was derived by Stokes [88] but to emphasize that it is valid only under the limit of $R_e \ll 1$ and in an unbounded medium. Since we cannot perform a computation on an infinite domain, we need to correct Eq. (101) to account for the presence of rigid walls. The effects caused by the rigid walls involve an increase in the viscous dissipation which decreases the speed of the sphere. In particular, the correction factor due to the wall-attachment effects K for creeping flow of a Newtonian fluid is defined by

$$K = \frac{v_{Stokes}}{v_T}. \tag{102}$$

It describes the decrease in the speed resulting from the presence of the walls as a function of the ratio of the radius of the falling sphere and a cylinder which contains the surrounding fluid. Being a the radius of the sphere and A the radius of the cylinder, Bohlin [89] obtains for $R_e \ll 1$ and $a/A < 0.6$,

$$K = \left[1 - 2.10443\left(\frac{a}{A}\right) + 2.08877\left(\frac{a}{A}\right)^3 + \cdots\right]^{-1}. \tag{103}$$

Experimental works [90] also obtain expressions which yield very similar results under the assumptions stated above. We can now compare the theoretical result with our numerical result. As in the previous section, the solid has to be

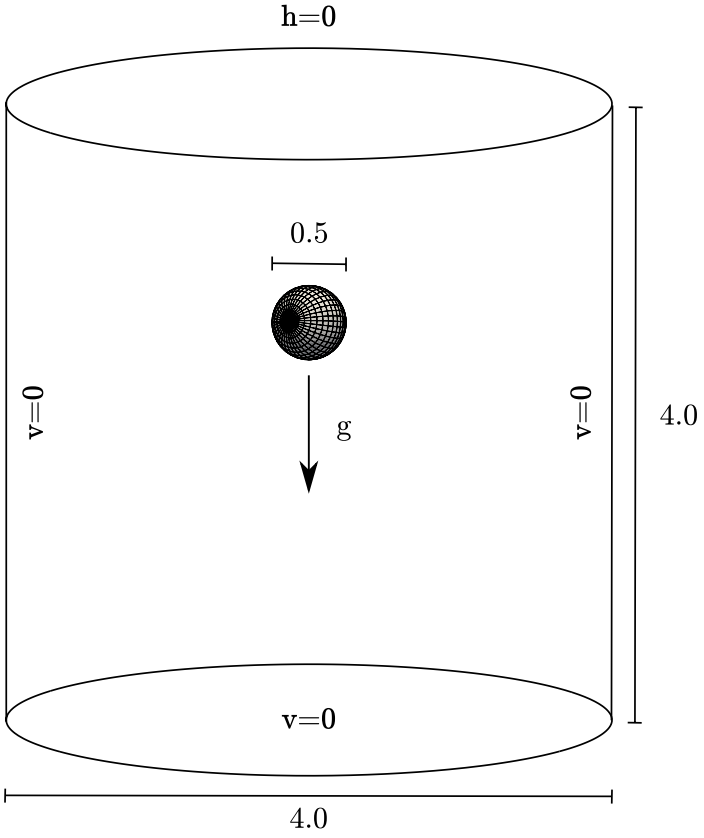


Fig. 8. Geometry, boundary conditions, and Lagrangian mesh for the free falling sphere. The lengths are in centimeters.

sufficiently stiff in order to complete the whole simulation without experiencing perceptible deformations and the length of the computational domain in the gravity direction needs to be long enough such that the sphere can reach its terminal velocity. The material used for the sphere is the same as that used for the cylinder in the previous section, but $d = 3$ in this case. The physical parameters of this example are the following: gravity acceleration $g = 981$ cm/s^2, dynamic viscosity $\mu = 10$ dyn/cm^2, fluid density $\rho^f = 1$ g/cm^3, solid density $\rho^s = 1.5$ g/cm^3 and shear modulus $\mu^s = 33\,550$ g/(cm s^2). The computational domain is a cylinder of 4 cm of height and $A = 2$ cm of radius. The radius of the sphere is $a = 0.25$ cm. The geometry of the problem, boundary conditions and Lagrangian mesh are represented in Fig. 8.

The Eulerian mesh has $60 \times 60 \times 67$ quadratic elements and it is nonuniform, as the mesh is finer through the path of the sphere. The geometry was generated by extruding a circle in the vertical (z) direction. To define the circle, we used a map that transforms each edge of the parameter space into a quarter of the circumference (details on the parameterization of the circle can be found, for example, in [91]). This makes the mapping singular at four points on the boundary, which produces four singular lines when the circle is extruded in vertical direction. Everywhere else, the basis functions are at least $\mathcal{C}^1$ continuous. The Lagrangian mesh has $9 \times 20 \times 40$ quadratic elements, the continuity between elements is $\mathcal{C}^1$ with the exception of three circles in which the continuity is $\mathcal{C}^0$ in an angular direction (this choice for the representation of a sphere has been used previously in FSI simulations [1]). NURBS shape functions, unlike Lagrange shape functions, are able to represent a sphere and a cylinder exactly when NURBS of second order or higher are used. The time step used in the simulation is $\Delta t = 10^{-3}$ s.

We have computed the average velocity in the gravity direction of the solid during the simulation. Again, the points chosen to calculate that average have been the images of the Greville points. As it can be seen in Fig. 9, the terminal average velocity of the simulation is very similar to the analytical result. In particular, the relative difference between the numerical and analytical terminal velocity is 3.2%. In Fig. 10(a), we plot the magnitude of the velocity, the streamlines and the Lagrangian mesh at time $t = 0.1$ s and in Fig. 10(b), we plot the velocity in z direction in two perpendicular slices at time $t = 0.1$ s together with the Eulerian mesh.

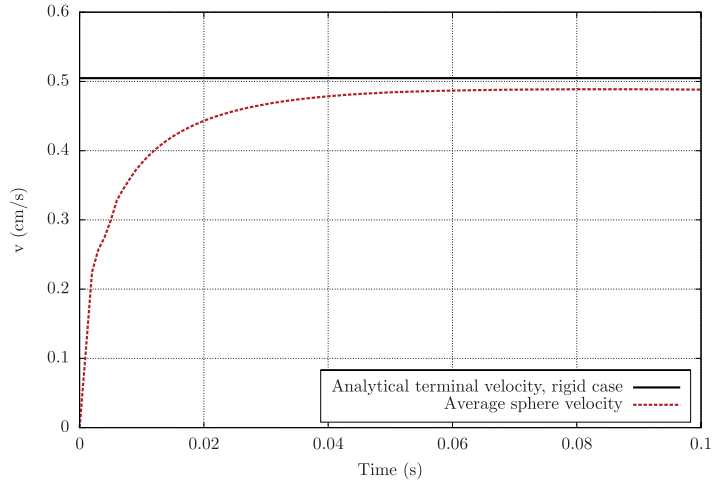
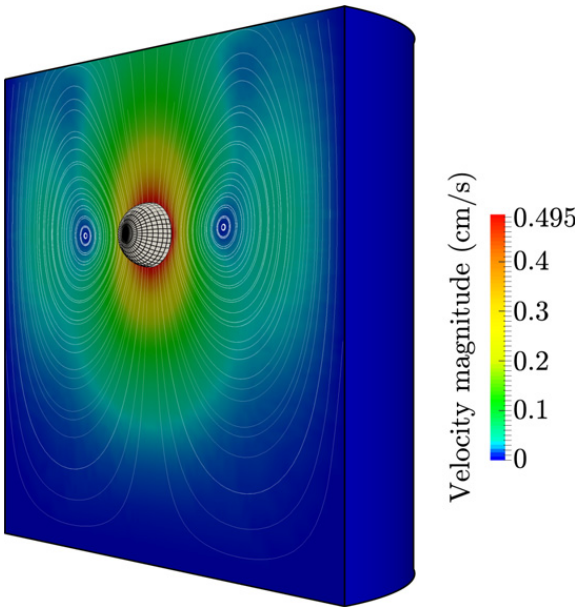
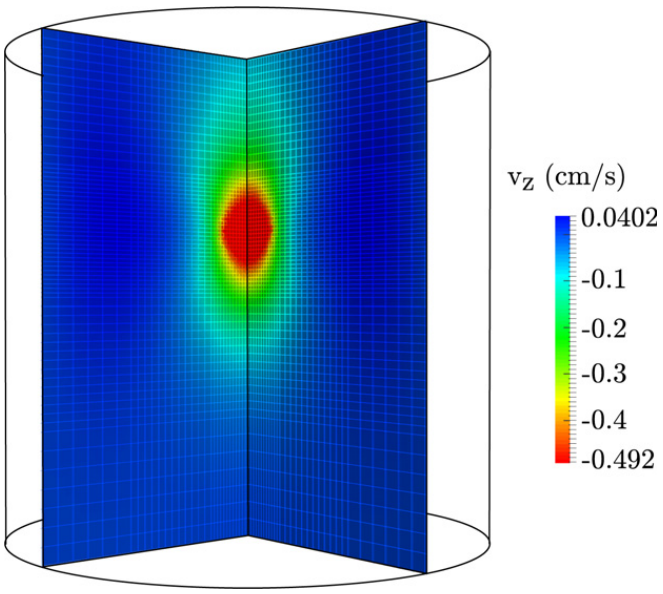


Fig. 9. (Color online) Analytical terminal velocity (black line) and time evolution of the average sphere velocity obtained from the computations. The Eulerian mesh is composed of $60 \times 60 \times 67$ elements and it is nonuniform, having higher spatial resolution around the sphere's trajectory (see Fig. 10(b)). The Lagrangian mesh has $9 \times 20 \times 40$ elements. We used quadratic basis functions.



(a) Velocity magnitude, $t = 0.1$ s.



(b) Vertical velocity, $t = 0.1$ s.

Fig. 10. (Color online) Contour plot of the velocity magnitude, streamlines, and Lagrangian mesh at time $t = 0.1$ s (a). Contour plot of the velocity in z (vertical) direction in two cut planes, along with the Eulerian mesh. On the Eulerian mesh, the size of the elements is smaller close to the sphere's trajectory (b).

4.3. *Parallelogram submerged in a flow*

In this case we consider a parallelogram embedded in a tube with Poiseuille-like flow. The same material model as in Section 4.1 is used, although a very low shear modulus is established so that the solid is very soft. All simulations are performed using the following physical parameters: gravity acceleration $g = 981$ cm/s^2, dynamic viscosity $\mu = 0.01$ dyn/cm^2, fluid density $\rho^f = 1$ g/cm^3 and solid density $\rho^s = 1.5$ g/cm^3.

The problem setup is shown in Fig. 11. We assume that for this problem the relevant Reynolds number is based on the inflow velocity (78 cm/s) and the side of the tube perpendicular to the flow (25 cm); its value is $R_e = 195\,000$. The initial position of the solid is such that its long side is perpendicular to the flow. The velocity gradient along the long side of the parallelogram will cause a rotation of the solid. During that process, it will experiment large deformations due to its low stiffness. As it will be shown, these large deformations can distort the Lagrangian mesh up to a point

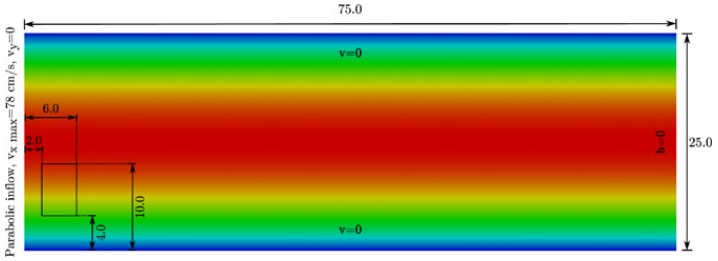
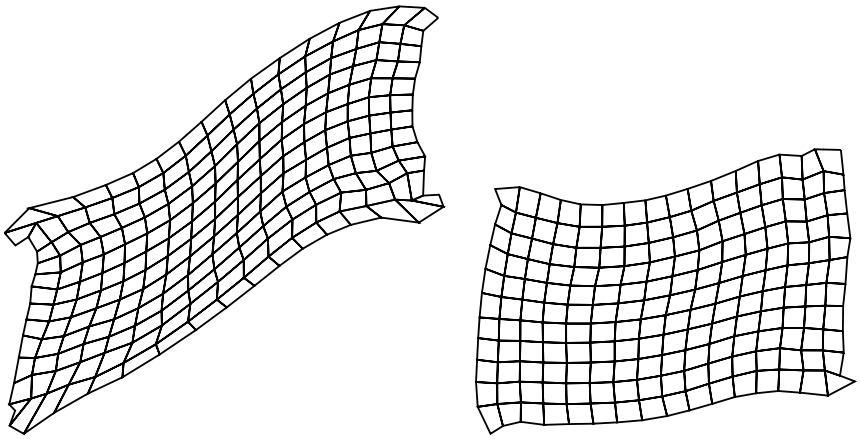


Fig. 11. Geometry, boundary conditions, initial position of the solid, and initial velocity for the parallelogram submerged in a flow. The color scale represents horizontal velocity with blue color corresponding to zero and red corresponding to 78 cm/s. The lengths are in centimeters. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) $\mathsf{p} = 1, t = 0.412$ s. (b) $\mathsf{p} = 2, t = 1.000$ s.

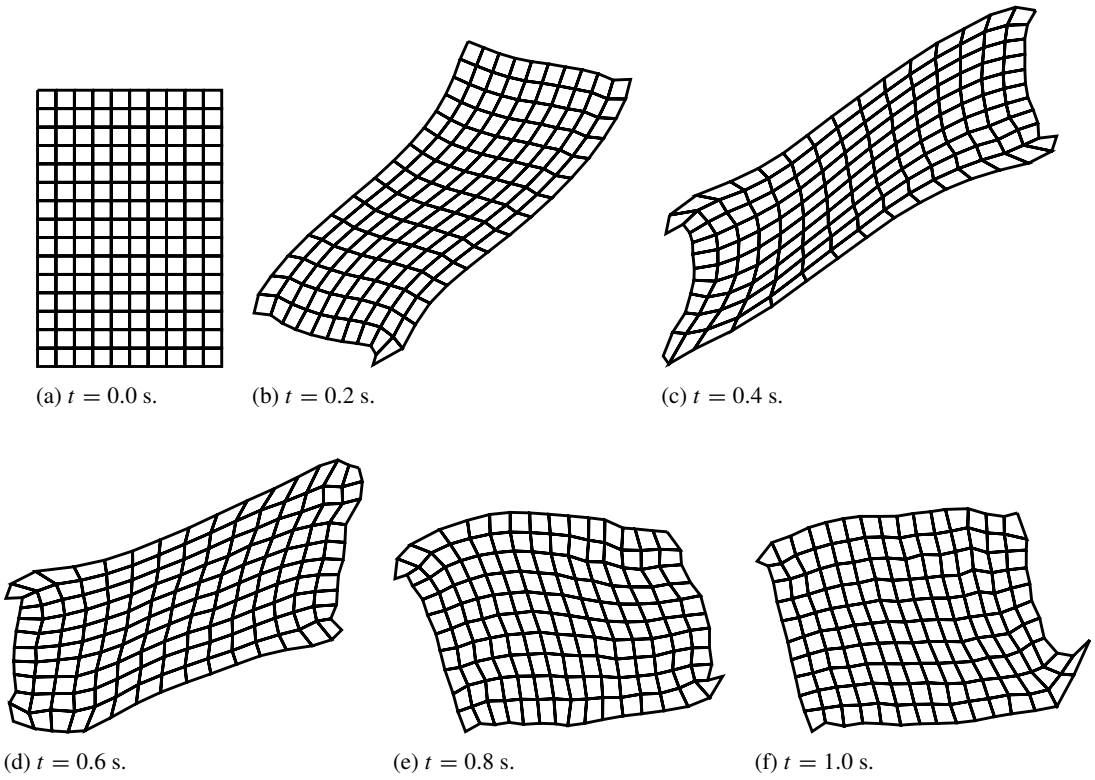
Fig. 12. Physical net of the Lagrangian mesh with 12×17 $\mathcal{C}^0$ linear elements a time step before the failure of the simulation (a), and physical net of the Lagrangian mesh with 11×16 C^1 quadratic elements when it has already completed the simulation (b). The shear modulus is $\mu^s = 400$ g/(cm s^2) and the number of control variables is 13×18 in both meshes.

where the simulation fails. In order to establish some criterion, we will consider the simulations to be successful if the solid is able to rotate 90°.

The time step used in the simulations is $\Delta t = 10^{-3}$ s. We will work with a fixed number of control variables in all the simulations of this section. In particular, 234 control variables. First, we take $\mu^s = 400$ g/(cm s^2). If we use a Lagrangian mesh with order $\mathsf{p} = 1$ and $\mathcal{C}^0$ continuity between elements, it collapses when the parallelogram has rotated approximately 45°. Nevertheless, a Lagrangian mesh with order $\mathsf{p} = 2$ and $\mathcal{C}^1$ continuity between elements is able to rotate 90° as shown in Fig. 12. We then decrease the shear modulus to $\mu^s = 300$ g/(cm s^2). The simulation fails for the two previous meshes with this new value of the shear modulus, but a Lagrangian mesh with order $\mathsf{p} = 3$ and $\mathcal{C}^2$ continuity between elements is able to complete the simulation without any problems as shown in Fig. 13. We are therefore able to deal with larger deformations as we increase the order of the mesh and the continuity between elements for the same number of degrees of freedom. The same fact was shown for isogeometric structural analysis in [36]. These results suggest that NURBS with higher order and higher continuity may increase the robustness of immersed FSI methods.

4.4. Hollow cylinders immersed in a flow

The fourth and final example consists of six hollow cylinders immersed in a water flow. The cylinders are filled with water and the problem setup is shown in Fig. 14. The inner radius of the cylinders is $r_{in} = 1.35$ cm, while the exterior radius is $r_{ex} = 1.5$ cm. The dimensions of the channel are 45×25 cm. The flow is driven by a parabolic inflow velocity with a maximum of 50 cm/s, the lateral boundaries are assigned zero velocity, and a zero-traction boundary condition is applied at the outflow. The initial condition for the velocity is a paraboloid with a maximum of 50 cm/s. We use the same solid material model as in Section 4.1. The shear modulus is $\mu^s = 335$ g/(cm s^2), which produces



(a) $t = 0.0$ s. (b) $t = 0.2$ s. (c) $t = 0.4$ s.

(d) $t = 0.6$ s. (e) $t = 0.8$ s. (f) $t = 1.0$ s.

Fig. 13. Physical net of the Lagrangian mesh with 10×15 $\mathcal{C}^2$ cubic elements at different time steps along the simulation. The shear modulus is $\mu^s = 300$ g/(cm s^2) and the number of control variables is 13×18.

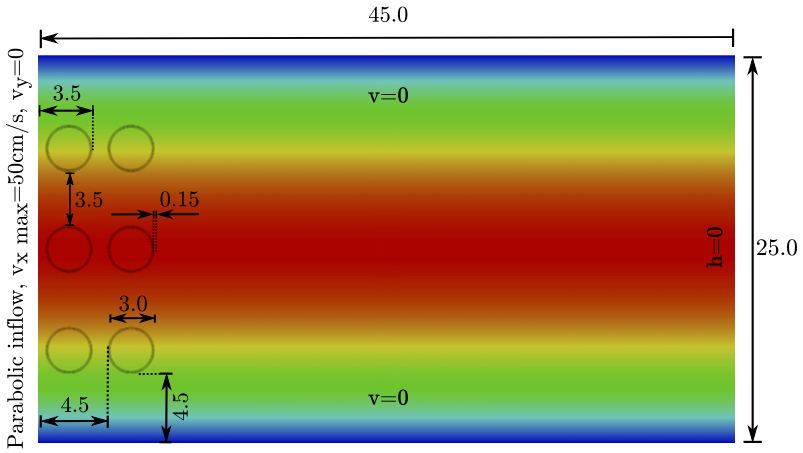


Fig. 14. Geometry, boundary conditions, initial position of the solids, and initial velocity for the 6 hollow cylinders immersed in a flow. The color scale represents horizontal velocity with blue color corresponding to zero and red corresponding to 50 cm/s. The lengths are in centimeters. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

highly deformable solids. We take $\rho^s = 1.5$ g/cm^3. The fluid is water with dynamic viscosity $\mu = 0.01$ dyn/cm^2 and density $\rho^f = 1.0$ g/cm^3. Defining the Reynolds number as in Section 4.3, we have $R_e = 125\,000$.

In this case, we use relatively thin structures for which we use solid elements with two $\mathcal{C}^1$ quadratic elements, that is, four $\mathcal{C}^1$ quadratic basis functions in the through-thickness direction. This way of modeling shell-like structures with solid NURBS elements was introduced in [26]. The Eulerian mesh has 210×175 quadratic elements and each Lagrangian mesh has 2×140 quadratic elements. The time step used in the simulation is $\Delta t = 10^{-3}$ s.

In Fig. 15 we plot the velocity magnitude and the pressure at different time steps where can see how the presence of the solids modifies the flow conditions. This example shows how the presence of thin solids can produce large pressure

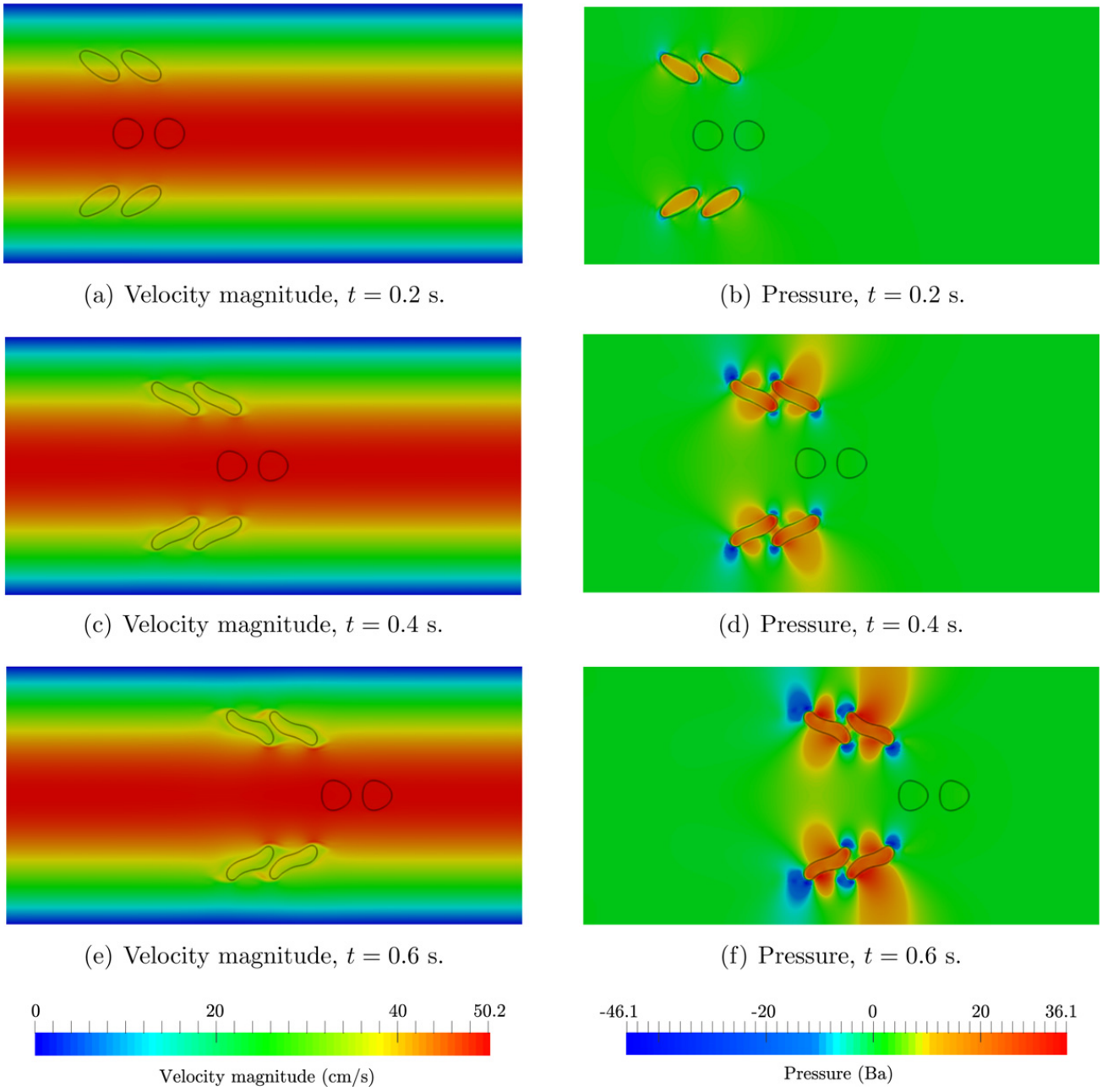


Fig. 15. (Color online) Contour plot of the velocity magnitude (left) and pressure (right) at different time steps. The simulation shows how the solid objects subjected to a velocity gradient experience large deformations, while those in the middle of the channel travel practically undeformed.

gradients that need to be resolved by the Eulerian mesh. Again, the ability of higher-order NURBS to approximate rough functions may increase the overall robustness of the algorithm.

5. Conclusions

We propose an immersed fluid–structure interaction method which uses NURBS-based isogeometric analysis for both the viscous incompressible fluid and the nonlinear hyperelastic solid. The high inter-element continuity of NURBS allows us to deal with larger deformations in the Lagrangian mesh, which increases the robustness of the overall algorithm. We are able to run very accurate simulations with the same element size in both meshes or even with bigger element size in the Lagrangian mesh, while works using standard C^0 finite elements consistently reported the need for a Lagrangian mesh at least twice as fine as the Eulerian mesh. The information transfer

between the Eulerian mesh and the Lagrangian mesh is kept to a minimum, and it is carried out by a collocation-like process that also exploits the high continuity of NURBS. We propose a fully-implicit, second-order accurate time integration algorithm based on the generalized-α method, which leads to a monolithic formulation. We validated our computational framework using theoretical data in two and three dimensions.

Acknowledgments

HG, HC and CB-C were partially supported by the European Research Council through the FP7 Ideas Starting Grant (project # 307201). HG was also partially supported by Xunta de Galicia, and by Ministerio de Economía y Competitividad (contract # DPI2013-44406-R). The last two grants are cofinanced with FEDER funds. This support is gratefully acknowledged. We acknowledge the open source scientific libraries PETSc, PetIGA, and their developers.

References

[1] Y. Bazilevs, K. Takizawa, T.E. Tezduyar, Computational Fluid–Structure Interaction: Methods and Applications, John Wiley & Sons, 2012.
[2] Y. Zhang, Y. Bazilevs, S. Goswami, C. Bajaj, T.J.R. Hughes, Patient-specific vascular NURBS modeling for isogeometric analysis of blood flow, Comput. Methods Appl. Mech. Engrg. 196 (2007) 2943–2959.
[3] Y. Bazilevs, M.-C. Hsu, Y. Zhang, W. Wang, T. Kvamsdal, S. Hentschel, J. Isaksen, Computational vascular fluid–structure interaction: methodology and application to cerebral aneurysms, Biomech. Model. Mechanobiol. 9 (2010) 481–498.
[4] K. Takizawa, T.E. Tezduyar, Computational methods for parachute fluid–structure interactions, Arch. Comput. Methods Eng. 19 (2012) 125–169.
[5] K. Takizawa, B. Henicke, T.E. Tezduyar, M.-C. Hsu, Y. Bazilevs, Stabilized space–time computation of wind-turbine rotor aerodynamics, Comput. Mech. 48 (2011) 333–344.
[6] W.K. Liu, Y. Liu, D. Farrell, L. Zhang, X. Wang, Y. Fukui, N. Patankar, Y. Zhang, C. Bajaj, J. Lee, J. Hong, X. Chen, H. Hsu, Immersed finite element method and its applications to biological systems, Comput. Methods Appl. Mech. Engrg. 195 (2006) 1722–1749.
[7] C.A. Felippa, K. Park, C. Farhat, Partitioned analysis of coupled mechanical systems, Comput. Methods Appl. Mech. Engrg. 190 (2001) 3247–3270.
[8] U. Küttler, W.A. Wall, Fixed-point fluid–structure interaction solvers with dynamic relaxation, Comput. Mech. 43 (2008) 61–72.
[9] E.H. Brummelen, P. Geuzaine, Fundamentals of Fluid–Structure Interaction, Wiley Online Library, 2010.
[10] S. Badia, A. Quaini, A. Quarteroni, Splitting methods based on algebraic factorization for fluid–structure interaction, SIAM J. Sci. Comput. 30 (2008) 1778–1805.
[11] J. Bueno, C. Bona-Casas, Y. Bazilevs, H. Gomez, Interaction of complex fluids and solids: theory, algorithms and application to phase-change-driven implosion, Comput. Mech. (2014). http://dx.doi.org/10.1007/s00466-014-1098-x. in press.
[12] J. Donéa, P. Fasoli-Stella, S. Giuliani, Lagrangian and Eulerian finite element techniques for transient fluid–structure interaction problems, in: Transactions of the 4th International Conference on Structural Mechanics in Reactor Technology Volume B: Thermal and Fluid/Structure Dynamics Analysis B(1/2), 1977, pp. 1–12.
[13] T. Belytschko, J.M. Kennedy, Computer models for subassembly simulation, Nucl. Eng. Des. 49 (1978) 17–38.
[14] T.J.R. Hughes, W.K. Liu, T.K. Zimmermann, Lagrangian–Eulerian finite element formulation for incompressible viscous flows, Comput. Methods Appl. Mech. Engrg. 29 (1981) 329–349.
[15] Y. Bazilevs, V.M. Calo, T.J.R. Hughes, Y. Zhang, Isogeometric fluid–structure interaction: theory, algorithms, and computations, Comput. Mech. 43 (2008) 3–37.
[16] C. Peskin, Flow patterns around heart valves: a numerical method, J. Comput. Phys. 10 (1972) 252–271.
[17] D. McQueen, C. Peskin, Computer-assisted design of butterfly bileaflet valves for the mitral position, Scand. J. Thorac. Cardiovasc. Surg. 19 (1985) 139–148.
[18] R. Beyer Jr., A computational model of the cochlea using the immersed boundary method, J. Comput. Phys. 98 (1992) 145–162.
[19] R. Dillon, L. Fauci, A. Fogelson, D. Gaver III, Modeling biofilm processes using the immersed boundary method, J. Comput. Phys. 129 (1996) 57–73.
[20] L. Fauci, C. Peskin, A computational model of aquatic animal locomotion, J. Comput. Phys. 77 (1988) 85–108.
[21] L. Zhang, A. Gerstenberger, X. Wang, W.K. Liu, Immersed finite element method, Comput. Methods Appl. Mech. Engrg. 193 (2004) 2051–2067.
[22] W.K. Liu, D. Kim, S. Tang, Mathematical foundations of the immersed finite element method, Comput. Mech. 39 (2007) 211–222.
[23] M. Gay, L. Zhang, W.K. Liu, Stent modeling using immersed finite element method, Comput. Methods Appl. Mech. Engrg. 195 (2006) 4358–4370.
[24] Y. Liu, L. Zhang, X. Wang, W.K. Liu, Coupling of Navier–Stokes equations with protein molecular dynamics and its application to hemodynamics, Internat. J. Numer. Methods Fluids 46 (2004) 1237–1252.
[25] Y. Liu, W.K. Liu, Rheology of red blood cell aggregation by computer simulation, J. Comput. Phys. 220 (2006) 139–154.
[26] T.J.R. Hughes, J.A. Cottrell, Y. Bazilevs, Isogeometric analysis CAD, finite elements, NURBS, exact geometry and mesh refinement, Comput. Methods Appl. Mech. Engrg. 194 (2005) 4135–4195.
[27] H. Gomez, V.M. Calo, Y. Bazilevs, T.J.R. Hughes, Isogeometric analysis of the Cahn–Hilliard phase-field model, Comput. Methods Appl. Mech. Engrg. 197 (2008) 4333–4352.

[28] H. Gomez, T.J. Hughes, X. Nogueira, V.M. Calo, Isogeometric analysis of the isothermal Navier–Stokes–Korteweg equations, Comput. Methods Appl. Mech. Engrg. 199 (2010) 1828–1840.
[29] H. Gomez, J. París, Numerical simulation of asymptotic states of the damped Kuramoto–Sivashinsky equation, Phys. Rev. E 83 (2011) 046702.
[30] H. Gomez, X. Nogueira, A new space–time discretization for the Swift–Hohenberg equation that strictly respects the Lyapunov functional, Commun. Nonlinear Sci. Numer. Simul. 17 (2012) 4930–4946.
[31] H. Gomez, X. Nogueira, An unconditionally energy-stable method for the phase field crystal equation, Comput. Methods Appl. Mech. Engrg. 249 (2012) 52–61.
[32] U. Thiele, A.J. Archer, M.J. Robbins, H. Gomez, E. Knobloch, Localized states in the conserved Swift–Hohenberg equation with cubic nonlinearity, Phys. Rev. E 87 (2013) 042915.
[33] H. Gomez, L. Cueto-Felgueroso, R. Juanes, Three-dimensional simulation of unstable gravity-driven infiltration of water into a porous medium, J. Comput. Phys. 238 (2013) 217–239.
[34] L. Dedè, M.J. Borden, T.J. Hughes, Isogeometric analysis for topology optimization with a phase field model, Arch. Comput. Methods Eng. 19 (2012) 427–465.
[35] J. Kiendl, K.-U. Bletzinger, J. Linhard, R. Wüchner, Isogeometric shell analysis with Kirchhoff–Love elements, Comput. Methods Appl. Mech. Engrg. 198 (2009) 3902–3914.
[36] S. Lipton, J. Evans, Y. Bazilevs, T. Elguedj, T.J.R. Hughes, Robustness of isogeometric structural discretizations under severe mesh distortion, Comput. Methods Appl. Mech. Engrg. 199 (2010) 357–373.
[37] I. Akkerman, Y. Bazilevs, V.M. Calo, T.J.R. Hughes, S. Hulshoff, The role of continuity in residual-based variational multiscale modeling of turbulence, Comput. Mech. 41 (2008) 371–378.
[38] J.A. Evans, Y. Bazilevs, I. Babuška, T.J. Hughes, n-widths, sup-infs, and optimality ratios for the k-version of the isogeometric finite element method, Comput. Methods Appl. Mech. Engrg. 198 (2009) 1726–1741.
[39] Y. Bazilevs, V.M. Calo, Y. Zhang, T.J.R. Hughes, Isogeometric fluid–structure interaction analysis with applications to arterial blood flow, Comput. Mech. 38 (2006) 310–322.
[40] Y. Bazilevs, V.M. Calo, J.A. Cottrell, T.J.R. Hughes, A. Reali, G. Scovazzi, Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows, Comput. Methods Appl. Mech. Engrg. 197 (2007) 173–201.
[41] J.A. Evans, T.J. Hughes, Isogeometric divergence-conforming b-splines for the unsteady Navier–Stokes equations, J. Comput. Phys. 241 (2013) 141–167.
[42] H. Gomez, T.J.R. Hughes, Provably unconditionally stable, second-order time-accurate, mixed variational methods for phase-field models, J. Comput. Phys. 230 (2011) 5310–5327.
[43] R. Dhote, H. Gomez, R. Melnik, J. Zu, Isogeometric analysis of a dynamic thermo-mechanical phase-field model applied to shape memory alloys, Comput. Mech. (2013) 1–16.
[44] G. Vilanova, I. Colominas, H. Gomez, Capillary networks in tumor angiogenesis: from discrete endothelial cells to phase-field averaged descriptions via isogeometric analysis, Int. J. Numer. Methods Biomed. Eng. 29 (2013) 1015–1037.
[45] G. Vilanova, I. Colominas, H. Gomez, Coupling of discrete random walks and continuous modeling for three-dimensional tumor-induced angiogenesis, Comput. Mech. (2013) 1–16.
[46] J. Liu, H. Gomez, J.A. Evans, T.J. Hughes, C.M. Landis, Functional entropy variables: a new methodology for deriving thermodynamically consistent algorithms for complex fluids, with particular reference to the isothermal Navier–Stokes–Korteweg equations, J. Comput. Phys. 248 (2013) 47–86.
[47] J.A. Cottrell, A. Reali, Y. Bazilevs, T.J.R. Hughes, Isogeometric analysis of structural vibrations, Comput. Methods Appl. Mech. Engrg. 195 (2006) 5257–5296.
[48] T.J. Hughes, A. Reali, G. Sangalli, Duality and unified analysis of discrete approximations in structural dynamics and wave propagation: comparison of p-method finite elements with k-method NURBS, Comput. Methods Appl. Mech. Engrg. 197 (49) (2008) 4104–4124.
[49] T. Elguedj, Y. Bazilevs, V.M. Calo, T.J. Hughes, $\overline{B}$ and $\overline{F}$ projection methods for nearly incompressible linear and non-linear elasticity and plasticity using higher-order NURBS elements, Comput. Methods Appl. Mech. Engrg. 197 (2008) 2732–2762.
[50] D. Benson, Y. Bazilevs, M. Hsu, T.J.R. Hughes, Isogeometric shell analysis: the Reissner–Mindlin shell, Comput. Methods Appl. Mech. Engrg. 199 (2010) 276–289.
[51] J. Kiendl, Y. Bazilevs, M.-C. Hsu, R. Wüchner, K.-U. Bletzinger, The bending strip method for isogeometric analysis of Kirchhoff–Love shell structures comprised of multiple patches, Comput. Methods Appl. Mech. Engrg. 199 (2010) 2403–2416.
[52] L. De Lorenzis, I. Temizer, P. Wriggers, G. Zavarise, A large deformation frictional contact formulation using NURBS-based isogeometric analysis, Internat. J. Numer. Methods Engrg. 87 (2011) 1278–1300.
[53] R. Dimitri, L. De Lorenzis, M. Scott, P. Wriggers, R. Taylor, G. Zavarise, Isogeometric large deformation frictionless contact using t-splines, Comput. Methods Appl. Mech. Engrg. 269 (2014) 394–414.
[54] R. Dimitri, L. De Lorenzis, P. Wriggers, G. Zavarise, NURBS-and t-spline-based isogeometric cohesive zone modeling of interface debonding, Comput. Mech. (2014) 1–20.
[55] W. Wall, M. Frenzel, C. Cyron, Isogeometric structural shape optimization, Comput. Methods Appl. Mech. Engrg. 197 (2008) 2976–2988.
[56] A. Buffa, G. Sangalli, R. Vázquez, Isogeometric analysis in electromagnetics: B-splines approximation, Comput. Methods Appl. Mech. Engrg. 199 (2010) 1143–1152.
[57] J.A. Cottrell, T.J.R. Hughes, Y. Bazilevs, Isogeometric Analysis Toward Integration of CAD and FEA, Wiley, 2009.
[58] C. Hesch, A. Gil, A. Arranz Carreño, J. Bonet, P. Betsch, A Mortar approach for fluid–structure interaction problems: immersed strategies for deformable and rigid bodies, Comput. Methods Appl. Mech. Engrg. 278 (2014) 853–882.
[59] T.J. Hughes, Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods, Comput. Methods Appl. Mech. Engrg. 127 (1995) 387–401.

[60] T.J.R. Hughes, G. Feijóo, L. Mazzei, J.-B. Quincy, The variational multiscale method—a paradigm for computational mechanics, Comput. Methods Appl. Mech. Engrg. 166 (1998) 3–24.
[61] T.J.R. Hughes, G. Sangalli, Variational multiscale analysis: the fine-scale Green's function, projection, optimization, localization, and stabilized methods, SIAM J. Numer. Anal. 45 (2007) 539–557.
[62] J. Chung, G. Hulbert, A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-method, J. Appl. Mech. 60 (1993) 371–375.
[63] K. Jansen, C. Whiting, G. Hulbert, Generalized-α method for integrating the filtered Navier–Stokes equations with a stabilized finite element method, Comput. Methods Appl. Mech. Engrg. 190 (2000) 305–319.
[64] A.N. Brooks, T.J. Hughes, Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations, Comput. Methods Appl. Mech. Engrg. 32 (1982) 199–259.
[65] J.E. Marsden, T.J. Hughes, Mathematical Foundations of Elasticity, Courier Dover Publications, 1994.
[66] P.-L. Lions, Mathematical Topics in Fluid Mechanics. Vol. 1. Incompressible Models, Oxford Science, 1996.
[67] M.-C. Hsu, D. Kamensky, Y. Bazilevs, M. Sacks, T. Hughes, Fluid–structure interaction analysis of bioprosthetic heart valves: significance of arterial wall deformation, Comput. Mech. 54 (2014) 1055–1071.
[68] D. Kamensky, M.-C. Hsu, D. Schillinger, J.A. Evans, A. Aggarwal, Y. Bazilevs, M.S. Sacks, T.J. Hughes, An immersogeometric variational framework for fluid–structure interaction: Application to bioprosthetic heart valves, Comput. Methods Appl. Mech. Engrg. 284 (2015) 1005–1053. Also available as ICES Report 14-12.
[69] J.C. Simo, T.J.R. Hughes, Computational Inelasticity, Springer, 2008.
[70] F. Auricchio, L. Da Veiga, T.J.R. Hughes, A. Reali, G. Sangalli, Isogeometric collocation methods, Math. Models Methods Appl. Sci. 20 (2010) 2075–2107.
[71] D. Schillinger, J. Evans, A. Reali, M. Scott, T.J.R. Hughes, Isogeometric collocation: cost comparison with Galerkin methods and extension to adaptive hierarchical NURBS discretizations, Comput. Methods Appl. Mech. Engrg. 267 (2013) 170–232.
[72] H. Gomez, A. Reali, G. Sangalli, Accurate, efficient, and (iso) geometrically flexible collocation methods for phase-field models, J. Comput. Phys. 262 (2014) 153–171.
[73] H. Gomez, A. Reali, An isogeometric collocation approach for Bernoulli-euler beams and Krichhoff plates, Comput. Methods Appl. Mech. Engrg. 284 (2015) 623–636.
[74] R. Codina, J. Principe, O. Guasch, S. Badia, Time dependent subscales in the stabilized finite element approximation of incompressible flow problems, Comput. Methods Appl. Mech. Engrg. 196 (2007) 2413–2430.
[75] C. De Boor, A Practical Guide to Splines, Vol. 27, Springer-Verlag, New York, 1978.
[76] K.E. Jansen, S.S. Collis, C. Whiting, F. Shaki, A better consistency for low-order stabilized finite element methods, Comput. Methods Appl. Mech. Engrg. 174 (1999) 153–170.
[77] C. Hesch, A. Gil, A. Arranz Carreño, J. Bonet, On continuum immersed strategies for fluid–structure interaction, Comput. Methods Appl. Mech. Engrg. 247–248 (2012) 51–64.
[78] Y. Saad, M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (1986) 856–869.
[79] T. Dupont, R.P. Kendall, H. Rachford Jr., An approximate factorization procedure for solving self-adjoint elliptic difference equations, SIAM J. Numer. Anal. 5 (1968) 559–573.
[80] T.A. Oliphant, An implicit, numerical method for solving two-dimensional time-dependent diffusion problems, Quart. Appl. Math. 19 (1961) 221–229.
[81] T.F. Chan, H.A. Van der Vorst, Approximate and Incomplete Factorizations, Springer, 1997.
[82] N. Collier, L. Dalcin, V.M. Calo, PetIGA: high-performance isogeometric analysis, preprint arXiv:1305.4452.
[83] M. Bernal, V.M. Calo, N. Collier, G. Espinosa, F. Fuentes, J. Mahecha, Isogeometric analysis of hyperelastic materials using PetIGA, Procedia Comput. Sci. 18 (2013) 1604–1613.
[84] P.A. Vignal, N. Collier, V.M. Calo, Phase field modeling using PetIGA, Procedia Comput. Sci. 18 (2013) 1614–1623.
[85] S. Balay, M.F. Adams, J. Brown, P. Brune, K. Buschelman, V. Eijkhout, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, K. Rupp, B.F. Smith, H. Zhang, PETSc Web Page, 2014. http://www.mcs.anl.gov/petsc.
[86] S. Balay, M.F. Adams, J. Brown, P. Brune, K. Buschelman, V. Eijkhout, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, K. Rupp, B.F. Smith, H. Zhang, PETSc users manual, Tech. Rep. ANL-95/11—Revision 3.4, Argonne National Laboratory, 2013.
[87] J. Happel, H. Brenner, Low Reynolds Number Hydrodynamics—with Special Applications to Particulate Media, Springer, 1983.
[88] G. Stokes, On the effect of internal friction of fluids on the motion of pendulums, Trans. Cambridge Phil. Soc. 9 (1851) 6–106.
[89] X. Bohlin, On the drag on a rigid sphere moving in a viscous fluid inside a cylindrical tube, Trans. Roy. Inst. Teck. 155 (1960) 1–63.
[90] A. Francis, Wall effect in falling ball method for viscosity, Physics 4 (1933) 403–406.
[91] A. Vuong, C. Heinrich, B. Simeon, ISOGAT: a 2D tutorial MATLAB code for isogeometric analysis, Comput. Aided Geom. Design 27 (2010) 644–655.

B Paper #2: "Isogeometric collocation using analysis-suitable T-splines of arbitrary degree"

Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 301 (2016) 164–186
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Isogeometric collocation using analysis-suitable T-splines of arbitrary degree

Hugo Casquero[a,*], Lei Liu[b], Yongjie Zhang[b], Alessandro Reali[c,d], Hector Gomez[a]

[a] *Departamento de Métodos Matemáticos, Universidade da Coruña, Campus de A Coruña, 15071, A Coruña, Spain*
[b] *Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA*
[c] *Department of Civil Engineering and Architecture, University of Pavia, via Ferrata 3, 27100, Pavia, Italy*
[d] *Technische Universität München – Institute for Advanced Study, Lichtenbergstraße 2a, 85748, Garching, Germany*

Received 11 September 2015; received in revised form 15 December 2015; accepted 16 December 2015
Available online 30 December 2015

Abstract

This paper deals with the use of analysis-suitable T-splines of arbitrary degree in combination with isogeometric collocation methods for the solution of second- and fourth-order boundary-value problems. In fact, analysis-suitable T-splines appear to be a particularly efficient locally refinable basis for isogeometric collocation, able to conserve the cost of only one point evaluation per degree of freedom typical of standard NURBS-based isogeometric collocation. Furthermore, T-splines allow to easily create highly non-uniform meshes without introducing elements with high aspect ratios; this makes it possible to avoid the numerical instabilities that may arise in the case of problems characterized by reduced regularity when Neumann boundary conditions are imposed in strong form and elements with high aspect ratio are used. The local refinement properties of T-splines can be also successfully exploited to approximate problems where point loads are applied. Finally, several numerical tests are herein presented in order to confirm all the above-mentioned features, as well as the good overall convergence properties of the combination of isogeometric collocation and analysis-suitable T-splines.

Keywords: Collocation; Analysis-suitable T-splines; Arbitrary-degree T-splines; Isogeometric analysis; Bézier extraction; Kirchhoff plates

1. Introduction

Isogeometric analysis (IGA) was first introduced ten years ago by [1] with the main aim of bridging Computer Aided Design (CAD) and Finite Element Analysis (FEA). The basic IGA concept, based on the isoparametric paradigm, consisted of adopting the same basis functions used for geometry representations in CAD systems – such as, e.g., Non-Uniform Rational B-Splines (NURBS) – for the approximation of field variables. The original goal was a cost-saving simplification of the typically expensive mesh generation and refinement processes required by standard FEA. In addition, thanks to the high-regularity properties of its basis functions, IGA showed a better accuracy

* Corresponding author.
E-mail address: hugo.casquero@udc.es (H. Casquero).

http://dx.doi.org/10.1016/j.cma.2015.12.014
0045-7825/

per-degree-of-freedom and an enhanced robustness with respect to standard FEA [2]. Such a superior behavior was exploited in a number of applications ranging from solids and structures (see, e.g., [3–9]) to fluids (see, e.g., [10–12]) and fluid–structure interaction (see, e.g., [13–17]). Moreover, the newly available higher regularity opened also the door to geometrically flexible discretizations of higher-order partial differential equations in primal form such as, e.g., phase-field models [18–20], Kirchhoff–Love shells [21,22], among others.

A well-known important issue of IGA is related to the development of efficient integration rules able to reduce the high array formation costs induced by standard Gaussian quadrature, in particular when higher-order approximations are employed. *Ad-hoc* quadrature rules were proposed by several authors (cf., among others, [23] and references therein), but the development of a general and effective solution for Galerkin-based IGA methods is still an open problem. In an attempt to address this issue taking full advantage of the special possibilities offered by IGA and in particular by the available higher regularity, isogeometric collocation (IGA-C) schemes were proposed in [24,25]. The main idea consisted of the discretization of the governing partial differential equations in strong form, within the isoparametric paradigm, reducing the number of evaluations needed for array formation to only one per degree of freedom. The aim was to optimize the computational cost still relying on IGA geometrical flexibility and accuracy.[1] Detailed comparisons with both IGA and FEA Galerkin-based approaches were carried out in [26], showing IGA-C advantages in terms of accuracy versus computational cost, in particular for higher-order approximation degrees. Since its introduction, many promising significant works on IGA-C were published in different fields, including phase-field modeling [27], contact [28,29], nonlinear elasticity [29], as well as several interesting studies in the context of structural elements [30–34]. A version of IGA-C based on generalized B-splines was recently proposed in [35].

It is important to remark that a fundamental ingredient towards the full success of any computational method is represented by its capabilities in terms of local refinement, and, clearly, IGA-C cannot be an exception. Accordingly, a locally refinable version of IGA-C, based on hierarchical splines, was introduced in [26]. The results proposed in that work are surely promising and convincing, but it has also to be noted that, to avoid linear dependence or instability problems, the concept of "weighted IGA-C" needed to be introduced, leading to a higher number of evaluations per degree of freedom in the transition regions between refinement levels.

The main goal of the present work is to introduce a new locally refinable version of IGA-C based on analysis-suitable T-splines (ASTS). In fact, ASTS-based IGA probably represents so far the most promising and studied locally refinable version of isogeometric analysis[2] and, as a consequence, exploring the potential guaranteed by the combination of IGA-C and T-splines looks like a natural way to go. In addition, the use of ASTS within IGA-C avoids the need of resorting to the above-mentioned weighted collocation concept needed in the case of hierarchical splines, thus conserving the cost of only one evaluation per degree of freedom typical of NURBS-based IGA-C.

We also highlight that one of the drawbacks of NURBS-based IGA-C, recently observed in [28], is the fact that spurious oscillations in the numerical solution may arise in the case of problems characterized by reduced regularity, when Neumann boundary conditions are imposed in strong form and elements with high aspect ratio are used. The adoption of T-splines may be of help in these situations as well, allowing to easily create highly non-uniform meshes without introducing elements with high aspect ratios, and thus avoiding the appearance of spurious instabilities.

Finally, another problem that, to our knowledge, has still to be tackled in the context of IGA-C is the application of point loads, which have to be modeled as approximate Dirac deltas. To be properly solved, such a problem clearly requires a strong local refinement around the load point of application, and this could be easily guaranteed by the use of ASTS.

For all the above reasons, the adoption of analysis-suitable T-splines of arbitrary degree within isogeometric collocation methods looks particularly attractive, and this work aims at revealing its potential for the solution of second-order and fourth-order boundary-value problems.

The paper is organized as follows. In Section 2, a detailed overview on T-splines and their analysis suitable version are given, while Section 3 is devoted to the description of isogeometric collocation methods for second-order and fourth-order problems. In Section 4, several numerical tests are presented to show the performance of the proposed combination of isogeometric collocation and analysis-suitable T-splines; these include four second-order problems characterized by different geometries and solutions that can be regular, singular, or presenting sharp layers or spikes,

[1] In general, IGA collocation features look particularly attractive when evaluation and formation costs are dominant, as in the case, e.g., of explicit structural dynamics [25].

[2] See Section 2 for an overview on T-splines, including the basic references on the topic.

as well as a fourth-order problem consisting of a simply supported circular Kirchhoff plate loaded by a point force. Finally, in Section 5, conclusions are drawn.

2. T-spline overview

In this section, we present a short overview of the T-spline technology. T-splines were originated in the design field [36–38] as a generalization of NURBS in order to circumvent some of the existent limitations of NURBS surfaces, namely, the lack of local refinement and the difficulty to create watertight surfaces in general cases. The amount of degrees of freedom needed in order to perform finite element analysis in an accurate way is typically much higher than the number of degrees of freedom that we need to represent exactly a certain geometric design. As a consequence, the geometrical model coming from a CAD program is generally refined previously to be used in isogeometric analysis, which makes even more important the ability to perform local refinement. Pursuing this aim, T-splines were recently brought to analysis [39]. Significant advances have been carried out since then: mathematical background of T-splines has been studied [40–43], several procedures to create volumetric T-splines have been designed [44–48], the conversion of a trimmed NURBS surface to an untrimmed T-spline surface was tackled in [49], the conversion of an unstructured quadrilateral mesh to a standard T-spline surface was explained in [50], and ASTS-based isogeometric analysis was applied to several problems in computational mechanics [51–60]. In the design realm, cubic T-splines are prominent, but arbitrary-degree T-splines have recently received attention [61]. Naturally, the need for flexibility in the order of the approximation becomes more pressing when design is to be integrated with analysis. Here, we describe analysis-suitable T-splines of arbitrary degree in two dimensions (based on their existing mathematical background [41,42]) and apply them to isogeometric collocation. The main difficulty stems from the fact that T-splines of even and odd degree are generated in a slightly different way. In our presentation, we assume that the reader is familiarized with NURBS-based isogeometric analysis [2] and the concept of Bézier extraction [62,63]. Under that assumption, this section is practically self-contained. For simplicity, we use the same degree in both directions, that is, we do not consider a different degree in each direction [41] or hybrid-degree weighted T-splines which are T-splines with local p-refinement capability [64]. Throughout the paper, the degree of the functions will be denoted by p.

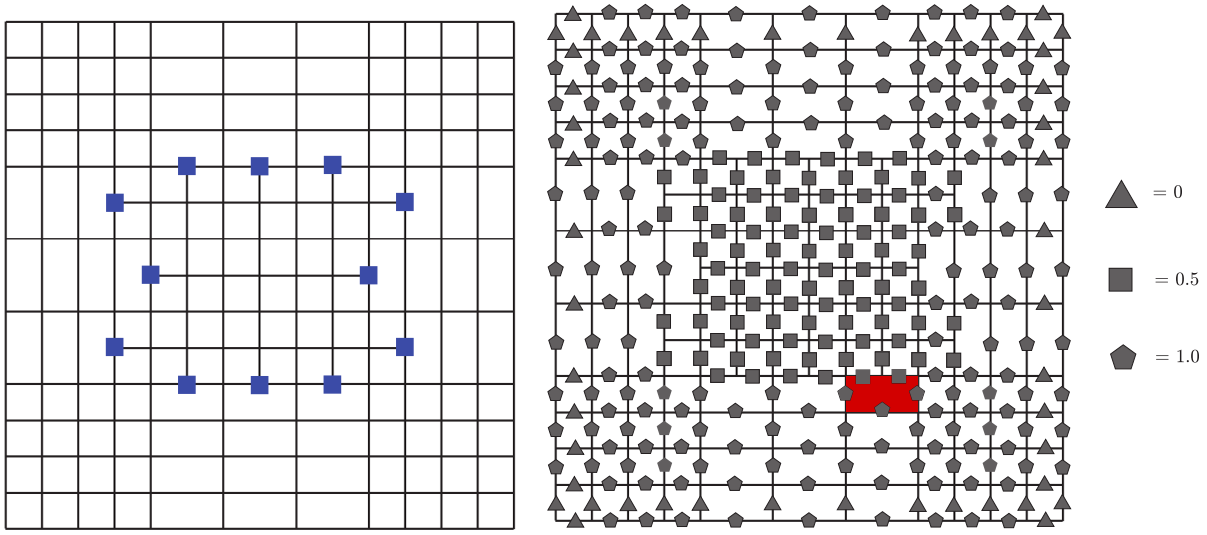
2.1. T-mesh in index space and knot interval configuration

The most fundamental object of the T-spline technology is the so-called *T-mesh in index space*. Although there are slightly different definitions of the T-mesh in index space in the literature, here we follow that given in [39]. When no confusion arises, we will simply use the term T-mesh *in lieu* of T-mesh in index space. Essentially, the T-mesh in index space is a polygonal tiling of a two-dimensional region, which encodes the topological information necessary to generate T-splines. Fig. 1(a) shows a sample T-mesh in index space. The polygons of the T-mesh are typically referred to as *faces*.[3] The corners of the faces are called *nodes* or *vertices*, and a line joining (exactly) two vertices is an *edge*. An essential difference of the T-mesh in index space with respect to its NURBS analogue is the presence of *T-junctions*, which are similar to hanging nodes in finite element analysis [see the blue squares in Fig. 1(a)]. Note that we exclude *ab initio* the presence of extraordinary nodes [65,53,66] in the T-mesh. To our knowledge, defining splines with $\mathcal{C}^2$ or higher continuity in the propinquity of an extraordinary node is still an open problem, which constitutes a significant obstacle for the application of our collocation methods.[4]

Once the concept of T-mesh has been introduced, our goal is to define spline functions over the T-mesh. These functions are called *T-spline blending functions*, and will be eventually used for analysis. To construct T-spline blending functions, we need to define a valid *knot interval configuration*. A knot interval configuration is defined by assigning a non-negative real number to each T-mesh edge. These non-negative real numbers are called *knot intervals* and may be interpreted as the difference of two adjacent knots [37]. We may also think of knot intervals as the length of the edges in a parametric space. We employ knot interval configurations which satisfy two conditions: (1) the knot intervals assigned to opposite sides of a given T-mesh face must sum to the same value; (2) the T-mesh has $\text{int}(p/2)$ outer rings of zero-length knot intervals, where $\text{int}(\cdot)$ represents the integer part of a real number. Fig. 1(b) shows a valid knot

[3] Occasionally, the word element is used in the literature as a synonym of face, but we will reserve it for something else.

[4] In [53], we consider a hybrid variational-collocation method for fluid–structure interaction which may be used on T-meshes with extraordinary nodes. The reason for this is that the algorithm is designed in a way that no derivatives need to be computed at the extraordinary nodes.



(a) T-mesh in index space. (b) Knot interval configuration.

Fig. 1. (Color online) (a) T-mesh in index space with twelve T-junctions marked with blue squares. (b) Valid knot interval configuration for the T-mesh represented in Fig. 1(a). The pentagons, squares and triangles correspond to knot intervals of 1, 1/2, and 0, respectively. This knot interval configuration is valid for $p = 2$ and $p = 3$, which are the cases used in this section to present the T-spline technology.

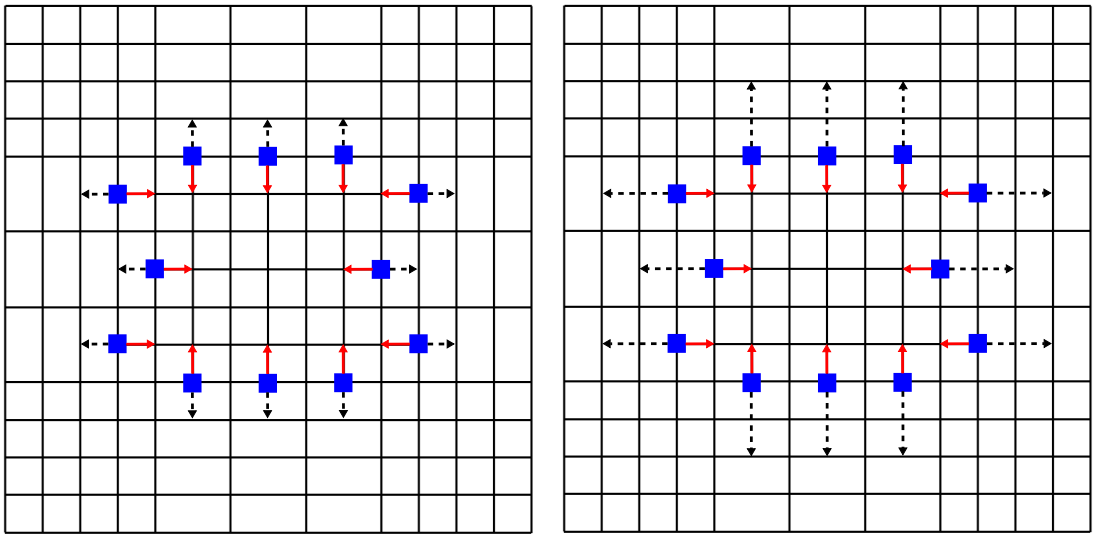
configuration of the T-mesh depicted in Fig. 1(a). Condition (1) is illustrated by the red face in Fig. 1(b). In particular, in the horizontal edges, we have one knot interval of value 1 in the bottom side and two knot intervals of value $1/2$ in the top side. The fulfillment of this condition guarantees that if the T-mesh is refined removing all T-junctions, there will be an underlying parametric space which is well defined. Condition (2) is convenient because it permits to infer easily the number of T-spline blending functions associated to a T-mesh. In particular, given a T-mesh, the global number of blending functions associated to it equals the number of vertices for odd-degree splines and the number of faces for even-degree functions.

Remark 2.1. We would like to mention that some authors define the T-mesh in index space and the knot interval configuration by setting p zero-length outer rings [41]. Although this definition has certainly some advantages, here we use the definition given in [39] to maintain the correspondence between the number of functions and the number of vertices (odd degree) or faces (even degree).

2.2. Extended T-mesh and elemental T-mesh

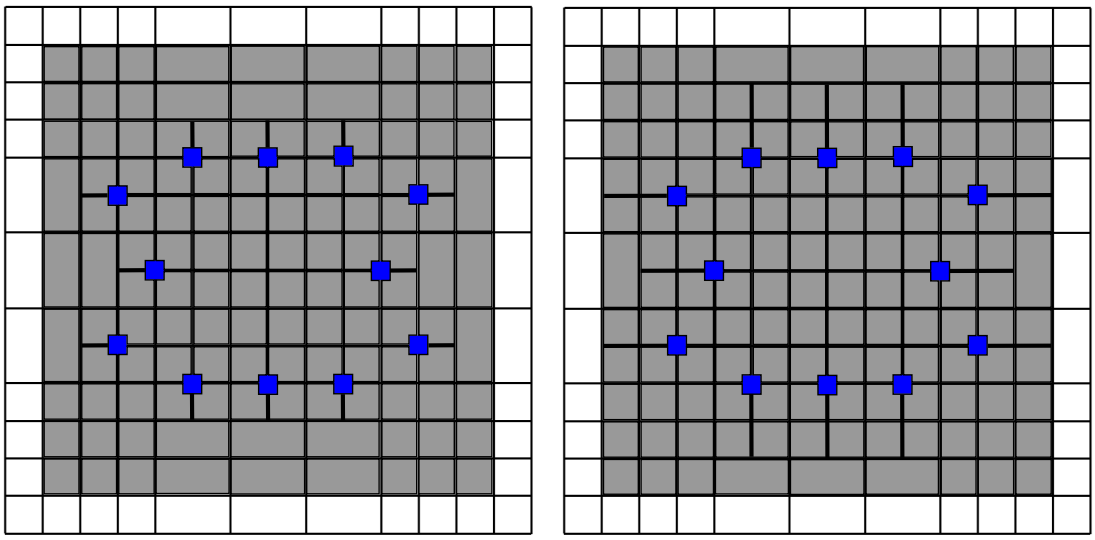
To define the extended T-mesh and the elemental T-mesh, we need to introduce first the concept of *T-junction extension*. A T-junction extension is composed of a face extension and an edge extension. A *face extension* is a directed line segment obtained as follows: we start at the T-junction and move in the direction of the missing edge until $\text{int}((p+1)/2)$ orthogonal edges are encountered. Face extensions are marked with black dashed arrows in Fig. 2 for $p = 2$ [panel (a)] and $p = 3$ [panel (b)]. We define an *edge extension* as a directed line segment obtained by moving in the opposite direction of the face extension until $\text{int}(p/2)$ orthogonal edges are encountered. Edge extensions are highlighted with red solid arrows in Fig. 2. When the T-junction extensions are plotted on top of the T-mesh, we get the *extended T-mesh*. The extended T-meshes for $p = 2$ and $p = 3$ corresponding to the T-mesh of Fig. 1(a) are shown in Fig. 2.

The *elemental T-mesh* is a new polygonal tiling that may be obtained from the extended T-mesh and the knot interval configuration. The polygons that form the elemental T-mesh will be called *elements*. T-mesh faces of zero area (i.e., zero measure) do not contribute elements to the elemental T-mesh and all T-mesh faces of non-zero area give rise to at least one element in the elemental T-mesh. Those which are not crossed by a T-junction extension give rise to exactly one element, and those crossed by T-junction extensions may give rise to more than one. Fig. 3 shows as shaded areas delimited by black lines the polygons that compose the elemental T-mesh. Note, in particular, how some T-mesh faces have been split into two elements in the elemental T-mesh. Notice also how the zero-area faces do not form elements (see the outer ring). The elements of the elemental T-mesh are important objects for analysis



(a) $p = 2$. (b) $p = 3$.

Fig. 2. Extended T-meshes associated to the T-mesh represented in Fig. 1(a) for $p = 2$ and $p = 3$. Face extensions are represented by black dashed lines and edge extensions are marked with solid red lines. Note that the red arrows hide the T-mesh edges located underneath them in the plot. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



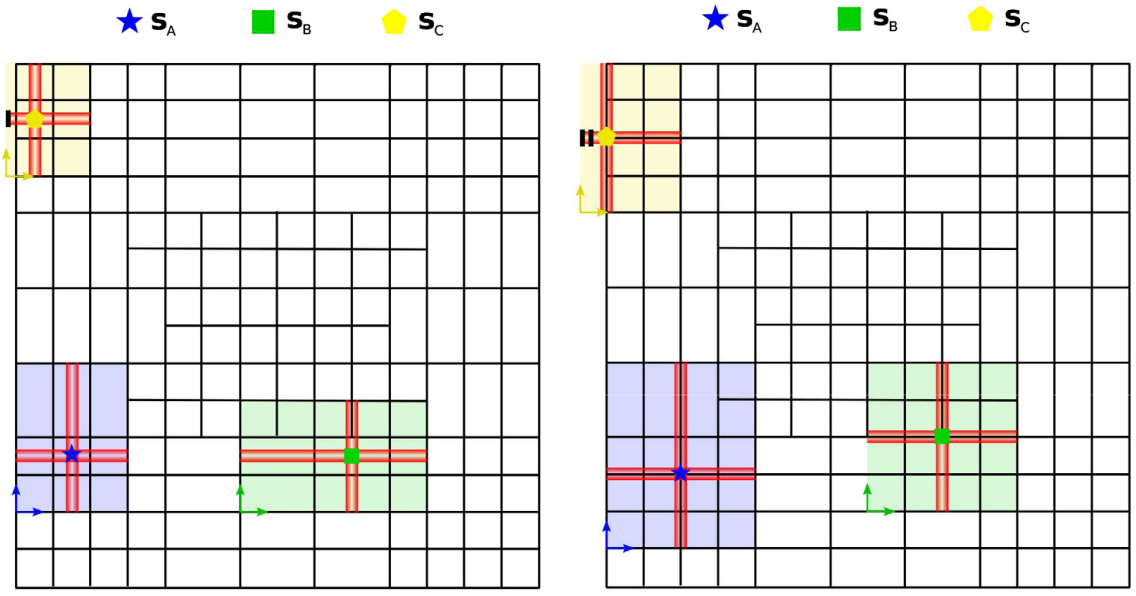
(a) $p = 2$. (b) $p = 3$.

Fig. 3. (Color online) Elemental T-meshes associated to the T-mesh represented in Fig. 1(a) for $p = 2$ and $p = 3$. The rectangles which are not shaded are not part of the elemental T-mesh. By comparing these plots with the extended T-meshes, it may be observed how faces of the T-mesh in index space have been split by face extensions giving rise to two elements in the elemental T-mesh.

because they delimit areas in which all T-spline blending functions are $\mathcal{C}^\infty$. The elements of the elemental T-mesh will be pushed forward to physical space using the isoparametric concept. Each element of the elemental T-mesh gives rise to a so-called Bézier element [63]. Bézier elements are suitable regions to perform numerical integration in an isogeometric analysis code based on variational forms.

2.3. Anchors, local knot interval vectors and local blending function coordinates

An *anchor* is simply a point of the index space to which we associate a T-spline blending function. The anchor associated to the Ath T-spline blending function will be denoted by $\mathbf{s}_A$. Note that here, and in what follows, indices denoted by capital letters take values of the global function numeration. Given a T-mesh in index space, the anchors are located at the vertices if p is odd and at the centers of the faces if p is even. Using the T-mesh in Fig. 1(a) as



(a) $p = 2$. (b) $p = 3$.

Fig. 4. (Color online) Anchors, local knot interval vectors, and local blending function coordinates associated to the T-mesh represented in Fig. 1(a) for $p = 2$ and $p = 3$. (a) The local knot interval vectors associated with the Ath, Bth, and Cth global blending functions for $p = 2$ are $\Delta_A\widehat{\Xi}^1 = \{0, 1, 1\}$, $\Delta_A\widehat{\Xi}^2 = \{1, 1, 1\}$, $\Delta_B\widehat{\Xi}^1 = \{1, 1, 1\}$, $\Delta_B\widehat{\Xi}^2 = \{1, 1, 1/2\}$, $\Delta_C\widehat{\Xi}^1 = \{0, 0, 1\}$, $\Delta_C\widehat{\Xi}^2 = \{1, 1, 0\}$. (b) The local knot interval vectors associated with the Ath, Bth, and Cth global blending functions for $p = 3$ are $\Delta_A\widehat{\Xi}^1 = \{0, 1, 1, 1\}$, $\Delta_A\widehat{\Xi}^2 = \{1, 1, 1, 1\}$, $\Delta_B\widehat{\Xi}^1 = \{1/2, 1/2, 1/2, 1\}$, $\Delta_B\widehat{\Xi}^2 = \{1, 1, 1/2, 1/2\}$, $\Delta_C\widehat{\Xi}^1 = \{0, 0, 0, 1\}$, $\Delta_C\widehat{\Xi}^2 = \{1, 1, 1, 0\}$.

our starting point, Fig. 4(a) and (b) shows the anchors associated to three particular blending functions with global indices A, B, and C, for even- and odd-degree T-splines, respectively. The location of the anchors is marked with a star, a square and a pentagon in the plot. Once we know the location of the anchor for a given function, we need to define a *local knot interval vector* in each direction of the index space. Let us denote by $\Delta_Q\widehat{\Xi}^i$ the local knot interval vector of the Qth function (or anchor) in the direction i, where $i = 1$ represents the horizontal direction and $i = 2$ the vertical one. The vector $\Delta_Q\widehat{\Xi}^i$ has length $p + 1$ for all Q and for all i, and may be written in component notation as $\Delta_Q\widehat{\Xi}^i = \{\Delta_Q\widehat{\xi}^i_1, \Delta_Q\widehat{\xi}^i_2, \ldots, \Delta_Q\widehat{\xi}^i_{p+1}\}$. A given component of the vector $\Delta_Q\widehat{\Xi}^i$ corresponds to the knot interval associated to a particular edge of the T-mesh that follows the direction i. Let us now explain how to compute the local knot interval vectors of a given anchor from the knot interval configuration. Put simply, the problem boils down to selecting a set of edges associated to the anchor and fill $\Delta_Q\widehat{\Xi}^i$ with their knot intervals. To identify those edges, we proceed as follows: Let us illustrate the process for $i = 1$, that is, let us define the vector $\Delta_Q\widehat{\Xi}^1$, which is associated to the horizontal direction. The procedure is identical for the vertical direction. The first step of the process is to place a horizontal segment centered in the anchor with sufficient length to cross exactly $p + 2$ orthogonal edges. Note that "centered in the anchor" means that the segment crosses the same number of orthogonal edges on the left- and right-hand sides of the anchor. This horizontal segment is represented in Fig. 4 by a thick, semi-transparent red line (for the time being, let us restrict our attention to the anchors $\mathbf{s}_A$ and $\mathbf{s}_B$). The fact that the segment crosses $p + 2$ orthogonal edges implies that it spans the length of a particular set of $p + 1$ edges. The knot intervals associated to those edges taken from left to right are the components of $\Delta_Q\widehat{\Xi}^1$. Note that condition (1) imposed to the knot interval configuration in Section 2.1 makes the choice of edges well defined. When an anchor is sufficiently close to the boundary to prevent us from defining a large enough segment so as to cross $p + 2$ orthogonal edges, we will add zero-length edges accordingly (see the anchors $\mathbf{s}_C$ in Fig. 4, which include thick black lines indicating the number of extra zero-length edges that need to be added). When we repeat this process in the vertical direction we obtain a cross centered in the anchor (defined by the horizontal and vertical red segments), which defines the support of the function in the obvious way. The supports are marked in Fig. 4 as rectangular colored areas. Using the bottom-left corner of each of these colored areas as origin, we can define the so-called *local blending function coordinates*. Note that there is one of these systems associated to each of the anchors and that they have been marked in Fig. 4 with arrows. Using as reference system the local blending function coordinates associated to a generic anchor $\mathbf{s}_Q$, the support of the Qth function, denoted $\widehat{\square}_Q$, is given by $\widehat{\square}_Q = [0, \sum_{j=1}^{p+1} \Delta_Q\widehat{\xi}^1_j] \times [0, \sum_{j=1}^{p+1} \Delta_Q\widehat{\xi}^2_j]$.

2.4. T-spline blending functions, Bézier extraction, and geometrical mapping

We proceed now to define the T-spline blending function associated to a generic anchor A. This function will be denoted by R_A. With the information given in Section 2.3, it is possible to anticipate which basis functions will have support on a given element e. To simplify notation, we introduce a local numbering for the blending functions as it is typically done in the finite element method. Here, we follow [67], and use the array IEN to establish a correspondence between local and global numbering. In particular, we use the formula $A = \text{IEN}(a, e)$, where A is a global blending function index, a is a local-to-element blending function counter, and e denotes the element number. Let us define the vector $\boldsymbol{N}^e = \{N_a^e\}_{a=1}^{n^e}$, where the functions N_a^e are progenitors of the blending functions with support on the element e. Note that, as emphasized with the superscript e in n^e, the length of the vector $\boldsymbol{N}^e$ changes from one element to another. This is a peculiar feature of T-splines which does not hold true for NURBS-based analysis. The concept of Bézier extraction permits computing $\boldsymbol{N}^e$ as a linear combination of the canonical tensor product Bernstein polynomials defined on a fixed parent element, namely, $\square = [-1, +1]^2$. This can be expressed mathematically as

$$\boldsymbol{N}^e(\boldsymbol{\xi}) = \boldsymbol{C}^e \boldsymbol{B}(\boldsymbol{\xi}) \quad \boldsymbol{\xi} \in \square, \tag{1}$$

where $\boldsymbol{B} = \{B_i^p\}_{i=1}^{(p+1)^2}$ is a vector containing the two-dimensional Bernstein polynomials of degree p in the domain $\square$ (see [65] for a precise description). The linear operator $\boldsymbol{C}^e$ can be represented by a rectangular matrix and it is called (element) Bézier extraction operator. Although we will not go into the details of the computation of $\boldsymbol{C}^e$, which may be found in [63], we point out that the Bézier extraction operator is computed from the knot interval configuration of the T-mesh using knot insertion.

In practical cases, it may be useful to utilize rational functions, typically to reproduce exactly a particular geometry. Given a set of global weights $\{w_A\}_{A=1}^n$ and their local counterparts for element e, namely, $\{w_a^e\}$, we can rationalize the functions in $\boldsymbol{N}^e$ as

$$R_a^e(\boldsymbol{\xi}) = \frac{w_a^e N_a^e(\boldsymbol{\xi})}{\sum_{b=1}^{n^e} w_b^e N_b^e(\boldsymbol{\xi})}, \tag{2}$$

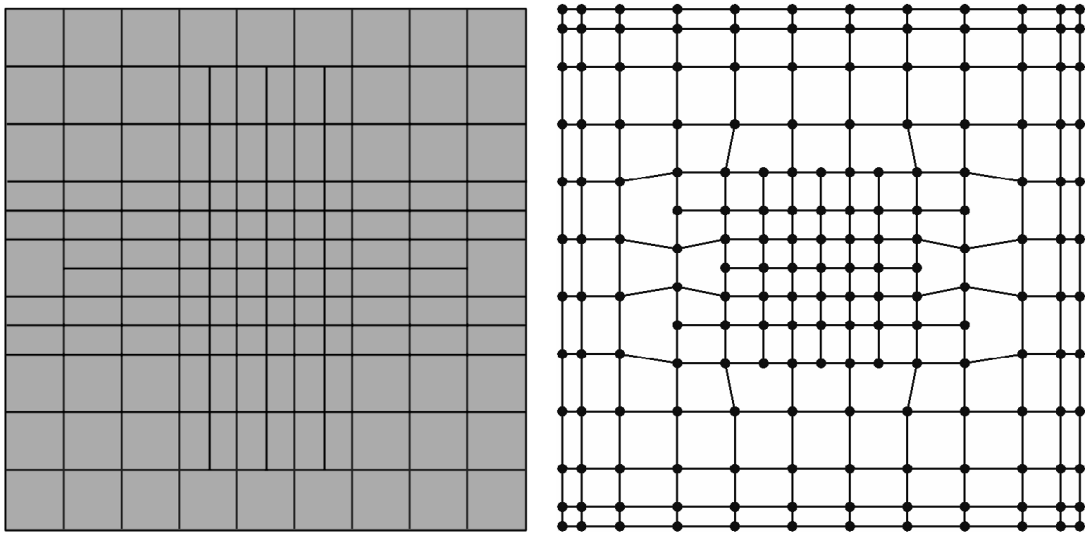
where R_a^e is the ath rational T-spline blending function over the element e. This finalizes the process of constructing the blending functions in parameter space.

Finally, to perform computations on non-trivial geometries, we need to map the parent element to physical space using a geometrical mapping $\boldsymbol{x}^e : \square \mapsto \Omega^e$, which goes from the parent element to the Bézier element Ω^e. The basic information needed to produce this mapping is a set of geometry control points $\{\boldsymbol{P}_B\}_{B=1}^n$ and the set of weights we just introduced, namely, $\{w_B\}_{B=1}^n$. A given geometry control point $\boldsymbol{P}_A$ will have local indices associated, such that $\boldsymbol{P}_A = \boldsymbol{P}_a^e$. The same argument applies to weights. Thus, the geometrical mapping local to element e can be defined as

$$\boldsymbol{x}^e(\boldsymbol{\xi}) = \sum_{a=1}^{n^e} \boldsymbol{P}_a^e R_a^e(\boldsymbol{\xi}) \quad \boldsymbol{\xi} \in \square. \tag{3}$$

Using the T-spline geometrical mapping given by Eq. (3), we can get the blending functions in physical space. In particular, the restriction of R_A to the element e is just the push forward of R_a^e. Eq. (3) will also be used to map the mesh to physical space. In summary, as shown by Eqs. (1)–(3), the use of Bézier extraction allows to use T-splines in a finite element code modifying just the shape function subroutine.

Remark 2.2. Fig. 5 shows the Bézier mesh and the control mesh associated to the T-mesh in Fig. 1(a) for a particular affine mapping with $p = 3$. The Bézier mesh could also be defined easily for $p = 2$ (or any other degree), but we do not report it here for conciseness. However, for $p = 2$ (even degree, in general) we do not generate the control meshes. For even-degree T-splines, due to the presence of T-junctions, there may be triangles or hexagons in the control mesh. The connectivity of the control points may be obtained from the dual T-mesh. A particular way to represent the control mesh of even degree T-splines is given in [61].



(a) Bézier mesh, $p = 3$. (b) Control mesh, $p = 3$.

Fig. 5. (Color online) Bézier mesh and control mesh associated to the T-mesh represented in Fig. 1(a) for a particular affine mapping with $p = 3$.

2.5. Analysis-suitable T-splines

If no topological constraints are imposed to the T-mesh, the resulting T-spline blending functions may not satisfy some relevant properties, such as, for example, linear independence or polynomial partition of unity. Analysis-suitable T-splines are a subset of T-splines which emanate from T-meshes satisfying a simple topological restriction, namely, no horizontal T-junction extension intersects a vertical T-junction extension [43,41]. ASTS retain the local refinement capabilities of T-splines, while satisfying all the important mathematical properties of NURBS. In particular, the blending functions associated to ASTS satisfy *a priori* the following properties:

(1) Polynomial partition of unity.
(2) Pointwise non-negativity.
(3) Linear independence.

Property (1) implies that affine transformations of a T-spline object in physical space may be produced by simply applying the transformation to the control points. Properties (1) and (2) guarantee that T-spline objects satisfy the convex-hull property. In [41,42], the authors also prove that ASTS are dual compatible, which implies optimal approximation properties.

3. Collocation algorithms for second- and fourth-order problems

In this section, we describe our algorithm for isogeometric collocation using ASTS. We focus on boundary-value problems, which involve second- and fourth-order partial-differential operators on mapped geometries. Collocation methods require the evaluation of the strong form of the equation at the so-called collocation points. Therefore, we need to compute the derivatives of the shape functions in physical space up to the maximum order attained in the PDE. Although straightforward, these computations are not standard in computational mechanics, so we include them in an Appendix for completeness.

3.1. Collocation method for second-order problems

We illustrate our collocation method for second-order PDEs using a classical example, namely, Poisson equation. Let us assume that Ω is an open set, representing the problem domain. The boundary of Ω is denoted as Γ. We call $\boldsymbol{n}$ the unit outward normal to Γ. We assume that Γ admits the decomposition $\Gamma = \overline{\Gamma_D \cup \Gamma_N}$ with $\overline{\Gamma_D \cap \Gamma_N} = \emptyset$, where $\emptyset$ denotes the empty set. To guarantee the well-posedness of the problem, we impose the constraint $\Gamma_D \neq \emptyset$,

while Γ_N is allowed to be the empty set. The problem can be formulated as: given $f : \Omega \mapsto \mathbb{R}$, $u_\Gamma : \Gamma_D \mapsto \mathbb{R}$ and $h_\Gamma : \Gamma_N \mapsto \mathbb{R}$, find $u : \Omega \mapsto \mathbb{R}$ such that,

$$\Delta u = f \qquad \text{in } \Omega, \tag{4}$$

$$u = u_\Gamma \qquad \text{on } \Gamma_D, \tag{5}$$

$$\nabla u \cdot \boldsymbol{n} = h_\Gamma \qquad \text{on } \Gamma_N. \tag{6}$$

Eqs. (5) and (6) represent the Dirichlet and Neumann boundary conditions, respectively. We assume that Ω can be represented by a T-spline geometrical object, which is mapped from a rectangular parameter space. To accommodate in our formulation typical numerical benchmarks (e.g., solutions on squares) and to fix ideas, we assume that the corners of the parameter space are mapped to points with discontinuous normal in physical space. We also allow for the data on the boundary to be discontinuous across the corners. Since the corner points will be collocation points for the boundary conditions in our algorithm for second-order PDEs, we need a procedure to define the normal and the boundary data at those points. Following the standard convention and the theory developed in [25], at those points we just take the arithmetic average of the data computed from the two intersecting boundary edges. In what follows, we will use the notation $\boldsymbol{n}^\star$ to refer to a modified normal which takes the value $\boldsymbol{n}$ everywhere except at the corners, where it has been redefined using the averaging procedure. Similarly, we use the notation $u^\star_\Gamma$ and $h^\star_\Gamma$.

Using the same geometrical mapping that we utilized to construct Ω, we can define a functional space in physical coordinates to approximate the solution to our problem. Let us call that space $\mathcal{V}^h = \text{span}\{R_A\}_{A=1}^n$. The R_A's are guaranteed to be linearly independent due to the use of ASTS, so $\dim(\mathcal{V}^h) = n$. Then, we can postulate the following approximation to the solution

$$u^h(\boldsymbol{x}) = \sum_{A=1}^{n} u_A R_A(\boldsymbol{x}), \tag{7}$$

which reduces our problem to finding the value of the u_A's for $A = 1, \ldots, n$. The strategy to compute the u_A's boils down to imposing that u^h satisfies the PDE or the boundary conditions at a total of n collocation points in physical space, which gives rise to a square system of equations. Note that we need u^h to be at least $\mathcal{C}^2$-continuous in the neighborhood of collocation points. The simplest way to satisfy this requirement is to take $p > 2$ in the algorithm and define a knot interval configuration that produces functions of maximum continuity. Let us call $\mathcal{M}^h = \{\boldsymbol{\tau}_B\}_{B=1}^n$ the set of collocation points in physical space. Then, the problem can be stated as: find $u^h(\boldsymbol{x}) = \sum_{A=1}^n u_A R_A(\boldsymbol{x})$ such that

$$\Delta u^h(\boldsymbol{\tau}_B) = f(\boldsymbol{\tau}_B) \qquad \forall \boldsymbol{\tau}_B \in \Omega, \tag{8}$$

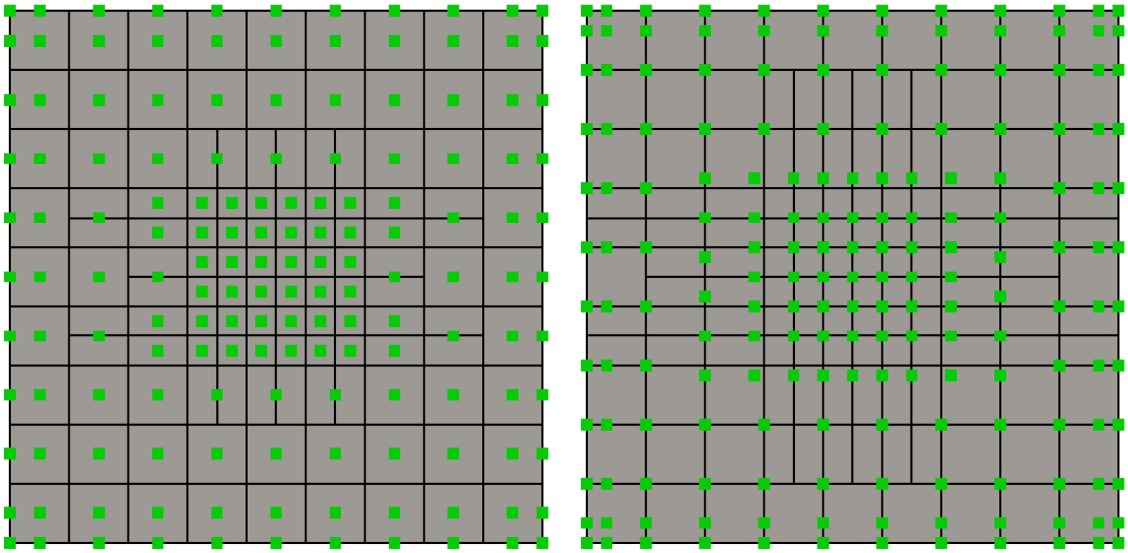
$$u^h(\boldsymbol{\tau}_B) = u^\star_\Gamma(\boldsymbol{\tau}_B) \qquad \forall \boldsymbol{\tau}_B \in \Gamma_D, \tag{9}$$

$$\nabla u^h(\boldsymbol{\tau}_B) \cdot \boldsymbol{n}^\star = h^\star_\Gamma(\boldsymbol{\tau}_B) \qquad \forall \boldsymbol{\tau}_B \in \Gamma_N. \tag{10}$$

To complete the description of our algorithm we just need to define the location of the collocation points, which controls the accuracy and stability of the method. In this article, we use Greville points as our collocation sites. Each T-spline basis function has a Greville point associated that can be obtained from its local knot interval vector. To define the Greville point associated to the Ath function, we initially define the location of the point using the local basis function coordinates defined in Section 2.3, and we use the usual definition of Greville points, which for a spline of degree p is just the average of p knot values (see [24]). Due to our choice to work with knot intervals, rather than knot values, which is customary in the T-spline community, we report here the formula that expresses the location of the Greville point in terms of the knot intervals, namely,

$$\widehat{\boldsymbol{\tau}}_A = \frac{1}{p}\sum_{k=1}^{p}(p+1-k)\begin{Bmatrix}\Delta_A\widehat{\xi}_k^1\\ \Delta_A\widehat{\xi}_k^2\end{Bmatrix}. \tag{11}$$

Once we know the location of the Ath Greville point in the local basis function domain, we need to find its coordinates in the parent element. The latter will be denoted by $\widetilde{\boldsymbol{\tau}}_A$. This is done in two steps using the classical affine mappings needed for Bézier extraction which are explained in detail in [63]. First, we use the affine mapping that goes from the local basis function coordinates to the domain of its corresponding Bézier element and then we use the mapping



(a) $p = 2$. (b) $p = 3$.

Fig. 6. (Color online) Bézier meshes associated to the T-mesh represented in Fig. 1(a) for a particular affine mapping with $p = 2$ and $p = 3$. The green squares denote the locations of Greville points in physical space.

that goes from the Bézier element to the parent element. The Greville points in physical space are computed using the element geometrical mapping as $\boldsymbol{\tau}_A = \boldsymbol{x}^e(\widetilde{\boldsymbol{\tau}}_A)$. Fig. 6 shows the Greville points in physical space for the T-mesh shown in Fig. 1(a) using $p = 2$ and $p = 3$ for a particular geometrical mapping. Note that due to our definition of the knot interval configuration and the local knot interval vectors, the T-spline blending functions with support in the vicinity of the boundary behave similarly to NURBS derived from open knot vectors. This naturally leads to some Greville points falling on the boundary. These collocation points are used to impose boundary conditions.

Remark 3.1. In isogeometric collocation methods, Neumann boundary conditions are usually collocated as shown in Eq. (10). Dirichlet boundary conditions, however, are normally built into the space strongly. Here, in order to simplify the algorithm, we decided to collocate Dirichlet boundary conditions as shown in Eq. (9). Both approaches are equivalent when we work with simple Dirichlet boundary conditions, such as for example, constants.

3.2. Collocation method for fourth-order problems

We illustrate our collocation method for fourth-order PDEs using a classical example, namely, the Kirchhoff plate model. As before, Ω and Γ denote the problem domain and boundary, respectively, and $\boldsymbol{n}$ the unit outward normal. We assume that the boundary Γ admits the decompositions $\Gamma = \overline{\Gamma_w \cup \Gamma_Q}$ and $\Gamma = \overline{\Gamma_\varphi \cup \Gamma_M}$ with $\overline{\Gamma_w \cap \Gamma_Q} = \emptyset$ and $\overline{\Gamma_\varphi \cap \Gamma_M} = \emptyset$. To ensure the well-posedness of the problem, we impose the constraint $\Gamma_w \neq \emptyset$. We formulate the following boundary-value problem over the spatial domain Ω as follows: given the functions $g : \Omega \mapsto \mathbb{R}$, $w_\Gamma : \Gamma_w \mapsto \mathbb{R}$, $\varphi_\Gamma : \Gamma_\varphi \mapsto \mathbb{R}$, $M_\Gamma : \Gamma_M \mapsto \mathbb{R}$, and $Q_\Gamma : \Gamma_Q \mapsto \mathbb{R}$, find $w : \Omega \mapsto \mathbb{R}$ such that,

$$D\Delta^2 w = g \qquad \text{in } \Omega, \tag{12}$$

$$w = w_\Gamma \qquad \text{on } \Gamma_w, \tag{13}$$

$$-\nabla w \cdot \boldsymbol{n} = \varphi_\Gamma \qquad \text{on } \Gamma_\varphi, \tag{14}$$

$$\nu D\Delta w + (1-\nu) D\boldsymbol{n} \cdot (\nabla\nabla w)\boldsymbol{n} = M_\Gamma \qquad \text{on } \Gamma_M, \tag{15}$$

$$D(\nabla(\Delta w) + (1-\nu)\,\boldsymbol{\Psi}(w)) \cdot \boldsymbol{n} = Q_\Gamma \qquad \text{on } \Gamma_Q. \tag{16}$$

Here, w is the deflection of the plate, $D = Et^3/(12(1-\nu^2))$ is the bending stiffness of the plate, ν is the Poisson ratio, E is the Young modulus, t is the thickness of the plate, g is the load per unit surface, w_Γ is the prescribed deflection,

φ_Γ is the prescribed rotation, M_Γ is the prescribed bending moment, Q_Γ is the prescribed effective shear which is a combination of the shear forces and the twisting moments, and $\boldsymbol{\Psi}(\cdot)$ is the following third-order operator

$$\boldsymbol{\Psi}(w) = \begin{Bmatrix} \dfrac{\partial^3 w}{\partial x \partial^2 y} \\ \dfrac{\partial^3 w}{\partial y \partial^2 x} \end{Bmatrix}. \tag{17}$$

As we did in Section 3.1, we use the Greville points as collocation points. Greville points in physical space are computed exactly in the same way. The collocation algorithm, however, is different. The difference stems from the fact that a well-posed fourth-order boundary value problem requires two boundary conditions at each point of the boundary. Unfortunately, if we imposed two boundary conditions at each Greville point that falls on the boundary and the PDE at all the interior Greville points, we would end up with more equations than unknowns. To circumvent this problem, a new algorithm was proposed in [30] for NURBS-based isogeometric collocation of Kirchhoff plates. Since such a recipe can be directly applied also in the case of ASTS-based collocation, it is the solution adopted herein. In the following we show the basic idea discussing a simple particular case; interested readers are referred to [30] for further details.

Despite the adopted collocation strategy is sufficiently general to accommodate any combination of boundary conditions represented by Eqs. (13)–(16), we illustrate it focusing on a simplified case where the same set of boundary conditions is applied on the entire boundary. Accordingly, we consider a simply supported plate, that is, $\Gamma_w = \Gamma_M = \Gamma$. Displacement boundary conditions ($w = 0$) can be equivalently strongly enforced in the space or collocated on all boundary collocation points. The problem is to properly collocate the remaining equations, i.e., the bending moment boundary conditions ($\nu D \Delta w + (1-\nu) D \boldsymbol{n} \cdot (\nabla\nabla w)\boldsymbol{n} = 0$) and the PDE ($D\Delta^2 w = g$), guaranteeing to have a number of equations equal to the number of unknowns. Our strategy is to collocate the PDE at all Greville points skipping the two outermost layers, while the bending moment boundary conditions are collocated only on boundary collocation points, following the additional criteria that corner points are skipped and the equations corresponding to the two closest points to a corner are averaged. In this way, the linear system of equations to be solved is square and our algorithm is completely described. To give a visual example of which points have to be used to collocate the PDE and the bending moment boundary conditions, we report in Fig. 7 a T-mesh in parametric space along with its Greville points. The black square Greville points are used to collocate the PDE, while the bending moment boundary conditions are collocated either on blue star or green pentagon points; the equations relative to green pentagon points adjacent to the same corner are then averaged. Note that red circle Greville points are not used to collocate the PDE or the bending moment boundary conditions.

4. Numerical results

In this section, we present five numerical examples that illustrate the performance of the proposed collocation methods based on analysis-suitable T-splines of arbitrary degree. First, we consider second-order elliptic problems with smooth and rough solution fields. We use these examples to investigate the numerical convergence rates of our algorithm. Then, we solve a common benchmark problem for the advection–diffusion equation. The solution exhibits boundary and internal layers where we use local refinement. We also consider a reaction–diffusion problem which develops spikes in the solution to illustrate the advantages of local refinement in this case. Finally, we apply our method to a Kirchhoff plate problem. Here, we take advantage of local refinement in order to capture the deformations caused by a point load in a mapped geometry. The code used to perform these simulations has been developed on top of the scientific library PETSc [68,69].

4.1. Second-order problem with smooth solution

We focus on the reaction–diffusion boundary-value problem

$$-\Delta u + u = f \qquad \text{in } \Omega, \tag{18}$$

$$u = 0 \qquad \text{on } \Gamma. \tag{19}$$

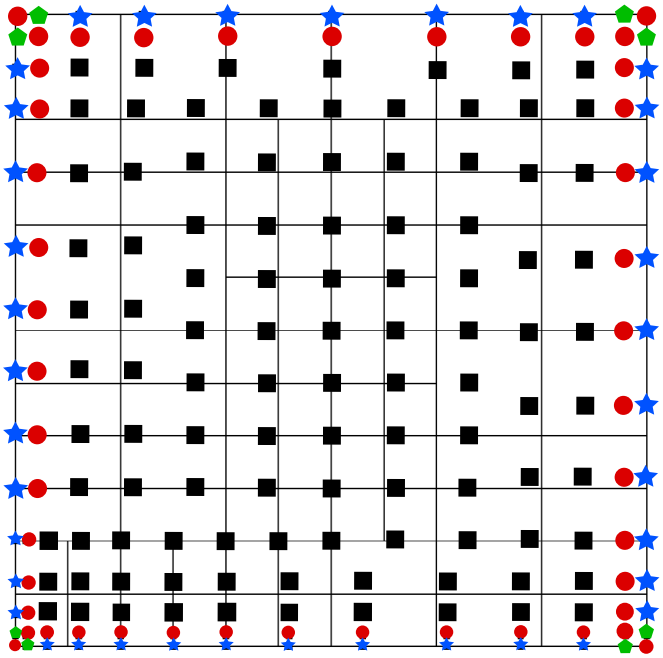


Fig. 7. (Color online) T-mesh in parametric space along with its Greville points. The black square Greville points are used to collocate the PDE, while the bending moment boundary conditions are collocated either on blue star or green pentagon points; the equations relative to green pentagon points adjacent to the same corner are then averaged. Note that red circle Greville points are not used to collocate the PDE or the bending moment boundary conditions.

The computational domain Ω is a quarter of an annulus situated in the positive quadrant of the Cartesian coordinate system. The inner radius R_i is equal to 1.0 and the outer radius R_o is equal to 4.0. We follow [26] and manufacture the source term f such that the exact solution to the problem is

$$u(x, y) = \theta(x, y)^2[\rho(x, y)^2 - 1][\rho(x, y)^2 - 16]\sin(x), \tag{20}$$

where $\rho(x, y) = \sqrt{x^2 + y^2}$ and $\theta(x, y) = \arctan(y/x)$ are the polar coordinates. Before we proceed further, let us denote the $L^2(\Omega)$ norm of a function u by $\|u\|_{L^2}$ and its H^1 seminorm by $|u|_{H^1}$. As usually, we take

$$\|u\|_{L^2} = \left(\int_\Omega u^2 \mathrm{d}\Omega\right)^{1/2} \quad \text{and} \quad |u|_{H^1} = \left(\int_\Omega |\nabla u|^2 \mathrm{d}\Omega\right)^{1/2}. \tag{21}$$

We define the error of the numerical solution as $e^h = u^h - u$. The $L^2(\Omega)$ norm and the $H^1(\Omega)$ seminorm of the relative error are defined as

$$r_{L^2} = \frac{\|e^h\|_{L^2}}{\|u\|_{L^2}} \quad \text{and} \quad r_{H^1} = \frac{|e^h|_{H^1}}{|u|_{H^1}}. \tag{22}$$

We solve this problem using quadratic, cubic, quartic, and quintic ASTS. We start considering a NURBS patch with 6×9 elements that represents the domain Ω exactly. Then, we perform quad-tree refinement using three levels of refinement in each mesh as it is illustrated in Fig. 8. We choose this type of refinement to be able to compare with the data in [26]. Fig. 8 also shows the numerical solutions for $p = 4$, which are indistinguishable at the scale of the plot. In Fig. 9 we report the numerical convergence rates of r_{L^2} and r_{H^1}. The plots show the evolution of the errors with respect to $N = \sqrt{n}$, which is the square root of the total number of degrees of freedom. The data suggest that the rates of convergence are of the type $\mathcal{O}(p)$ for even orders and $\mathcal{O}(p - 1)$ for odd degrees. These results are consistent with those reported in [26].

Remark 4.1. 1. One of the main advantages of NURBS-based isogeometric collocation is its efficiency, due to the fact that only one point evaluation per degree of freedom is needed. However, such an efficiency may be limited in many practical applications because NURBS lack local refinement capabilities. As anticipated in the introduction,

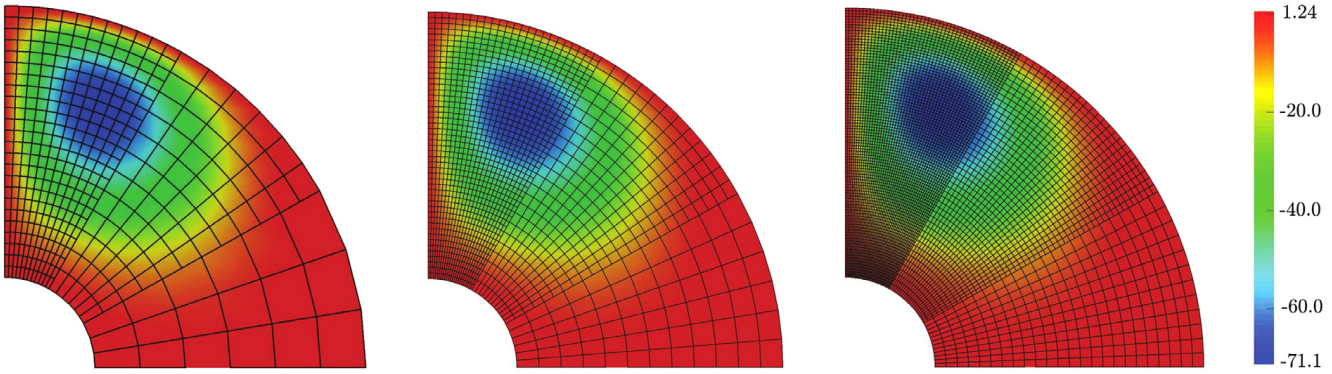


Fig. 8. (Color online) Second-order problem with smooth solution. Numerical solution on three different meshes using $p = 4$. The solutions are indistinguishable at the scale of the plot.

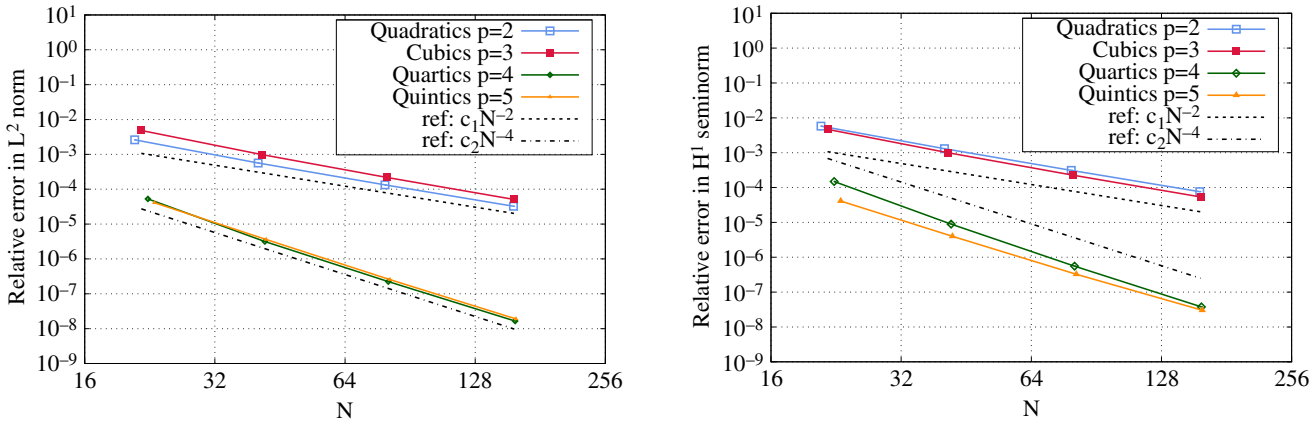


Fig. 9. (Color online) Numerical convergence rates of r_{L^2} and r_{H^1}. The data suggest that the algorithm converges to the exact solution with rates $\mathcal{O}(p)$ and $\mathcal{O}(p-1)$ for even and odd degrees respectively.

in [26], isogeometric collocation was extended to hierarchical NURBS. This allowed local refinement, but the one point evaluation per degree of freedom was lost in the regions between levels of refinement due to the use of weighted collocation. The generalization of isogeometric collocation to ASTS combines the advantages of the two previous methodologies, allowing local refinement but still guaranteeing only one point evaluation per degree of freedom.

2. As highlighted earlier, IGA-C requires u^h to be at least $\mathcal{C}^2$-continuous in the neighborhood of collocation points, and the simplest way to satisfy this requirement is to take $p > 2$ and define a knot interval configuration that produces functions of maximum continuity. However, extensive numerical testing shows that also the $\mathcal{C}^1$ quadratic case appears to give good results in all situations,[5] which is a significant practical result.

4.2. Second-order problem with rough solution

Here, we solve the Laplace equation on an L-shaped domain. More specifically, the problem is solved on $\Omega = [-1, 1]^2 \setminus ([0, 1] \times [-1, 0])$, as depicted in the left panel of Fig. 10. The problem can be stated as

$$\Delta u = 0 \qquad \text{in } \Omega, \tag{23}$$

$$u = u_\Gamma \qquad \text{on } \Gamma_D, \tag{24}$$

$$\nabla u \cdot \boldsymbol{n} = h_\Gamma \qquad \text{on } \Gamma_N, \tag{25}$$

[5] Clearly, in correspondence of a collocation point of $\mathcal{C}^1$-regularity, a single value has to be assigned to the discontinuous second derivative (e.g., the average of the different corresponding derivative values).

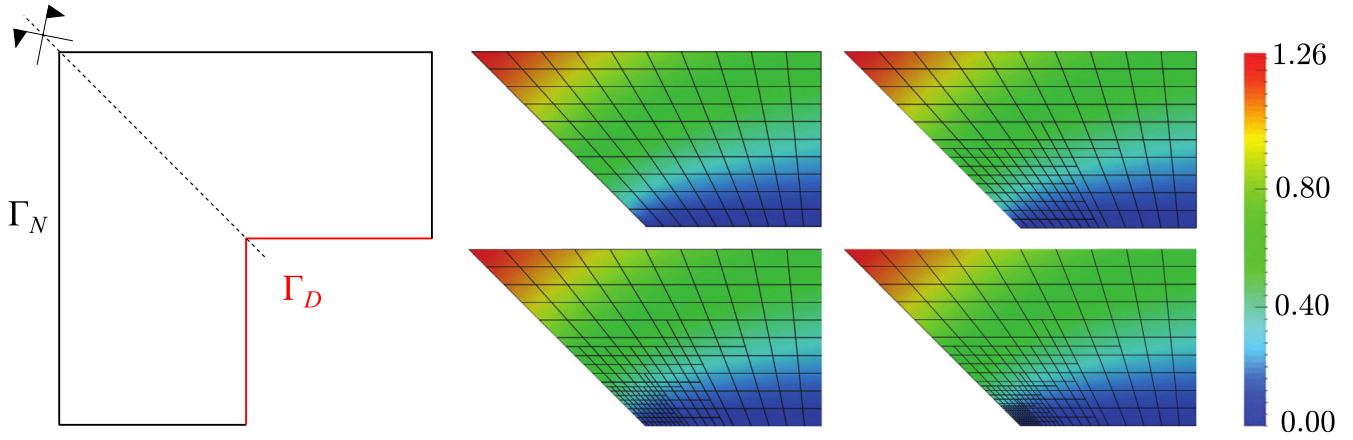


Fig. 10. (Color online) Problem domain and boundary decomposition showing the symmetry of the problem (left). Making use of the symmetry, we compute the numerical solution on a uniform NURBS mesh and several locally-refined T-spline grids using $p = 2$ (middle and right panels).

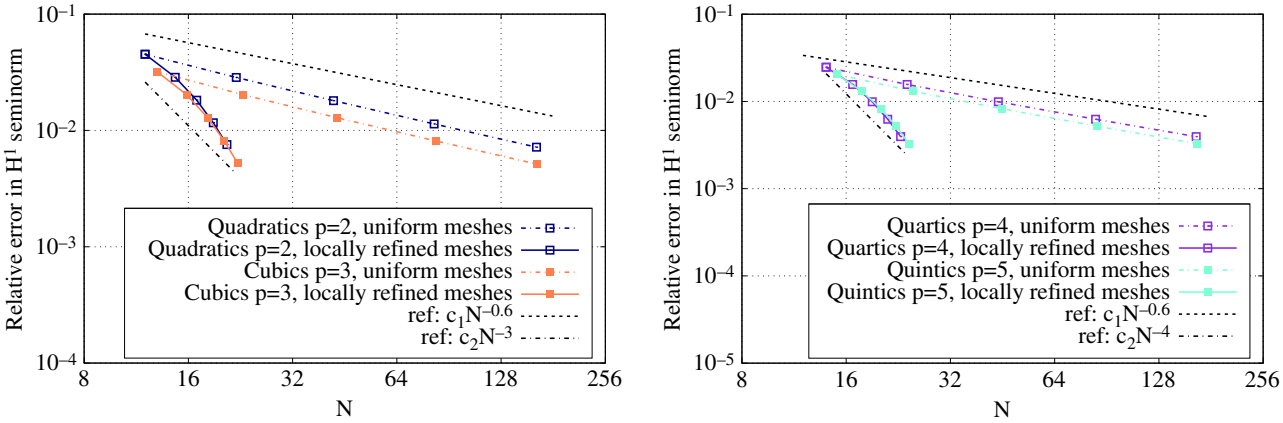


Fig. 11. (Color online) Numerical rates of convergence of r_{H^1} using uniform (dashed lines) and local refinement (solid lines). With uniform refinement the convergence rate stalls at approximately 0.6 for all p. Local refinement produces significantly higher apparent convergence rates.

where Γ_D is composed by the re-entrant boundaries and $\Gamma_N = \Gamma \setminus \Gamma_D$ (cf. the left panel of Fig. 10). The functions u_Γ and h_Γ are defined in such a way that the exact solution is

$$u(x, y) = \rho(x, y)^{\frac{2}{3}} \sin\left(\frac{2\theta(x, y)}{3}\right), \tag{26}$$

where ρ and θ denote again the usual polar coordinates. It may be observed that the first partial derivatives of u, which can be expressed as,

$$\frac{\partial u}{\partial x} = -\frac{2\sin(\theta/3)}{3\sqrt[3]{\rho}}; \qquad \frac{\partial u}{\partial y} = \frac{2\cos(\theta/3)}{3\sqrt[3]{\rho}} \tag{27}$$

tend to infinity as ρ tends to zero. Therefore, the exact solution of the problem is not smooth and this is expected to create a barrier in the convergence rate if uniform refinement is utilized. It is known, however, that local refinement can help alleviate this problem. The goal of this example is to show how our algorithm performs in this situation.

In our computations, we take advantage of the existing symmetry in the problem and work with half of the domain as indicated on the top of the middle panel of Fig. 10. Our initial mesh is a NURBS patch with 10×10 elements which is plotted along with the corresponding numerical solution in Fig. 10. Then, we subsequently add several levels of refinement in order to have enhanced resolution close to the singularity as shown in the remaining panels of Fig. 10. The solutions are indistinguishable at the scale of the plot, but the numerical convergence rates shown in Fig. 11 are revealing. Here, we plot the evolution of r_{H^1} (see Eq. (22)) as a function of $N = \sqrt{n}$ for quadratic, cubic, quartic, and quintic ASTS. We show data for uniformly (dashed lines) and locally (solid lines) refined meshes. For uniform

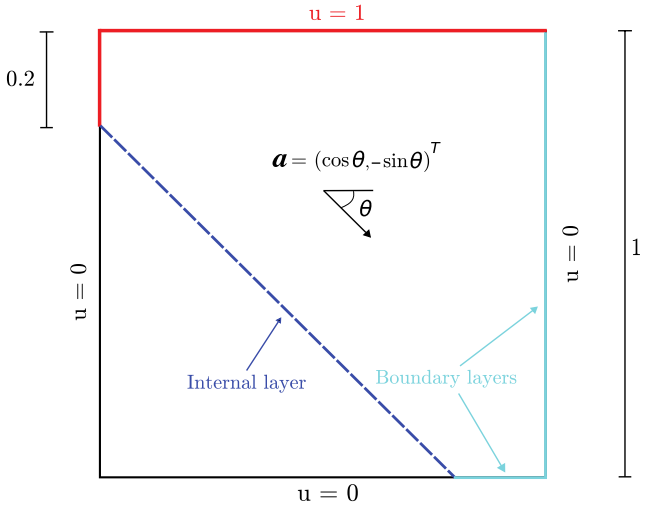


Fig. 12. (Color online) Problem description and boundary conditions for the advection–diffusion benchmark example.

refinement the rate of convergence stalls at approximately $\mathcal{O}(0.6)$ irrespectively of p due to the lack of smoothness of the solution. Local refinement allows to achieve an apparent convergence rate which is significantly higher.

Remark 4.2. In [28], it was reported the appearance of spurious oscillations in the numerical solution if Neumann boundary conditions are collocated in the standard way when the problem is characterized by a reduced regularity and elements have very high aspect ratios. Two alternatives were proposed in [28] in order to fix this issue, the so-called hybrid collocation and enhanced collocation methods. However, those techniques introduce small drawbacks in the overall algorithm, namely, hybrid collocation decreases the efficiency since the computation of boundary integrals is needed and enhanced collocation requires to determine a user-defined constant. Within NURBS-based IGA-C, the presence of elements with high aspect ratios is unavoidable in many practical situations, since refinement is always global for a NURBS patch. Instead, the local refinement properties of ASTS allow T-junctions making it possible to avoid elements with high aspect ratios. For example, if the local refinement is done as in Fig. 10, all elements have aspect ratio equal to one, except for a few elements with aspect ratio equal to two due to T-junction extensions. Therefore, as the current example clearly shows, the use of ASTS allows to collocate Neumann boundary conditions in the standard way without giving rise to any spurious oscillation in the numerical solution, also in the case of reduced regularity problems and highly locally refined meshes.

4.3. Advection–diffusion with boundary and internal layers

This section analyzes a classical advection–diffusion example typically referred to as advection skew to the mesh. The boundary-value problem can be written as

$$\boldsymbol{a} \cdot \nabla u - \nabla \cdot (\kappa \nabla u) = 0 \qquad \text{in } \Omega, \tag{28}$$

$$u = u_\Gamma \qquad \text{on } \Gamma, \tag{29}$$

where $\boldsymbol{a}$ denotes the (constant) velocity vector and κ the diffusion coefficient. The velocity field as well as the computational domain and u_Γ are depicted in Fig. 12. The angle of the velocity field is $\theta = \pi/4$. This is a classical benchmark in the literature of fluid mechanics and stabilized methods [70–72]. However, we do not focus on stabilization here, but on local refinement. The solution to the problem is controlled by a dimensionless quantity called Péclet number, which is defined as

$$Pe = \frac{|\boldsymbol{a}|L}{\kappa}, \tag{30}$$

where L is a length scale of the problem. We take $L = 1$, which is the side length of the problem domain. The Péclet number controls the relative strength of advection with respect to diffusion. It is known from theoretical developments that, for large Pe, the diffusion operator is relevant only in thin layers characterized by the length scale $\sim\kappa/|\boldsymbol{a}|$. Therefore, for large Péclet number we expect sharp layers in the solution emanating from the discontinuity points in the

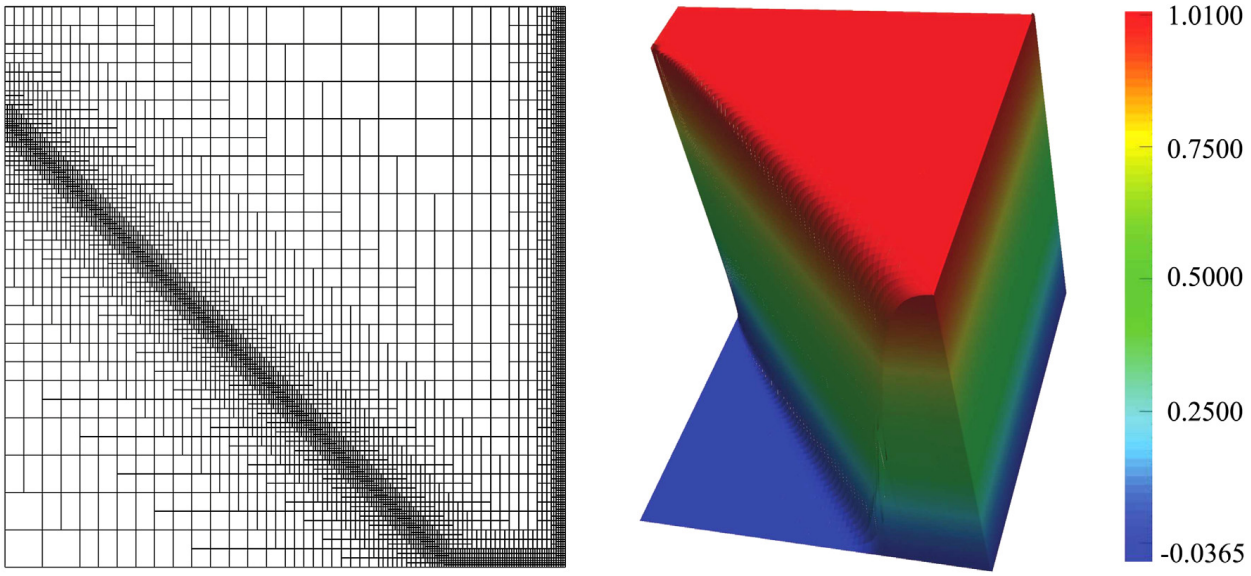


Fig. 13. Bézier mesh (left) and solution field (right) for the advection skew to the mesh problem. The Bézier mesh has five levels of refinement and gives rise to a total number of degrees of freedom $n = 5492$. The solution field is plotted with a color scale that ranges from the minimum to the maximum of the numerical solution showing that the usual undershoots and overshoots encountered in this problem are rather small. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

boundary data as depicted in Fig. 12. Unless such layers are properly resolved by the mesh or a suitable stabilization is introduced, the numerical solution is expected to present spurious oscillations.

We solve this problem for $Pe = 10^3$ using cubic ASTS-based collocation, without introducing any stabilization term. We locally refine the mesh close to the boundary and internal layers, using five levels of refinement as shown on the left hand side of Fig. 13. The total number of degrees of freedom in the space that emanates from the mesh in $n = 5492$. The right hand side of Fig. 13 shows the numerical solution using our algorithm. The color scale goes from the minimum to the maximum value of the numerical solution. It can be observed that the usual undershoots and overshoots of the numerical solution are rather small with our algorithm, which shows the effectiveness of the approach.

4.4. Reaction–diffusion with spikes

The boundary-value problem is defined by

$$cu - \nabla \cdot (\kappa \nabla u) = 0 \qquad \text{in } \Omega, \tag{31}$$

$$u = u_\Gamma \qquad \text{on } \Gamma, \tag{32}$$

where the computational domain and u_Γ are depicted in Fig. 14 (top left panel). The problem parameters are the diffusion coefficient κ and the reaction rate c, but dimensional analysis can be used to show that the solution depends only on the Damköhler number

$$Da = \frac{cL^2}{\kappa}. \tag{33}$$

Here, $L = 1$ is the side length of Ω. For large values of Da the solution is expected to be zero everywhere except close to the corners, where it progressively takes larger values until it reaches one as prescribed by the boundary conditions. We take the value $Da = 10^3$ in our simulations, following the studies on the same problem of [39]. We solve this problem using a quadratic ASTS mesh with seven levels of refinement in order to accurately capture the corner phenomena. The analysis results are plotted in the bottom panels of Fig. 14, where we only show a small region of the domain delimited in the top right panel. It can be seen that the solution appears to be stable even in the vicinity of very large gradients.

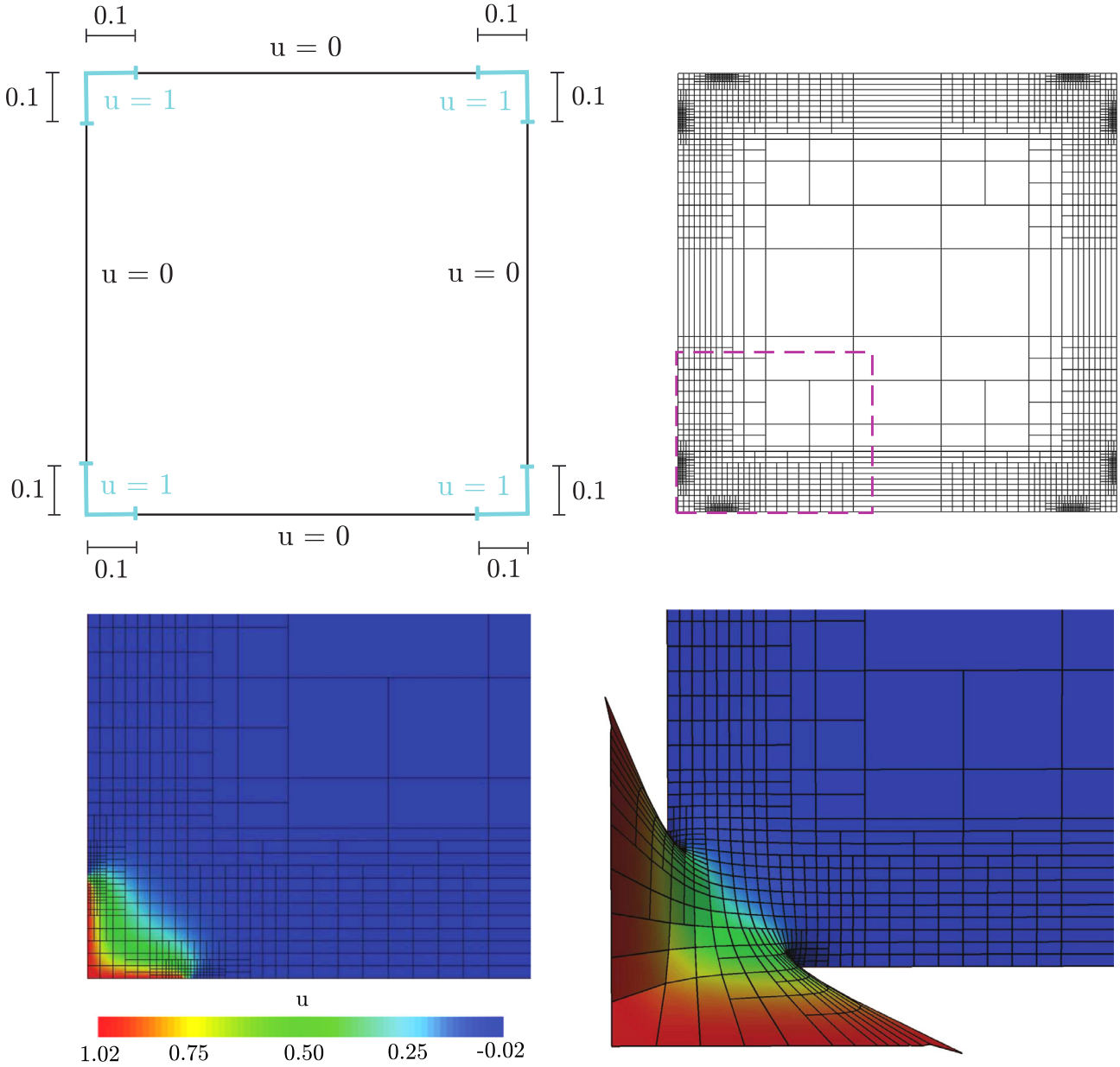


Fig. 14. From left to right and top to bottom, we present a panel showing the problem setup, the computational domain and the imposed boundary conditions for the reaction–diffusion example with spikes. Then, we show the Bézier mesh ($n = 2563$) where we have used seven levels of refinement. In the Bézier mesh we mark a small area with a dashed magenta line. The bottom panels show this area zoomed in. On the left hand side, we plot the solution using a color scale that goes from the minimum to the maximum of the numerical solution. The solution looks stable even in the vicinity of very large gradients. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

4.5. Circular Kirchhoff plate with a point load

Kirchhoff plates with distributed loads were solved using NURBS-based collocation methods in [30]. Here, we solve Kirchhoff plates with point loads using the local refinement capability of ASTS in order to refine close to the load.[6] A simply supported circular plate with radius $r = 1$ is considered. We apply a point load $P = 1$ to the

[6] We remark that, to our knowledge, this is the first time that isogeometric collocation is used to approximate problems with point loads.

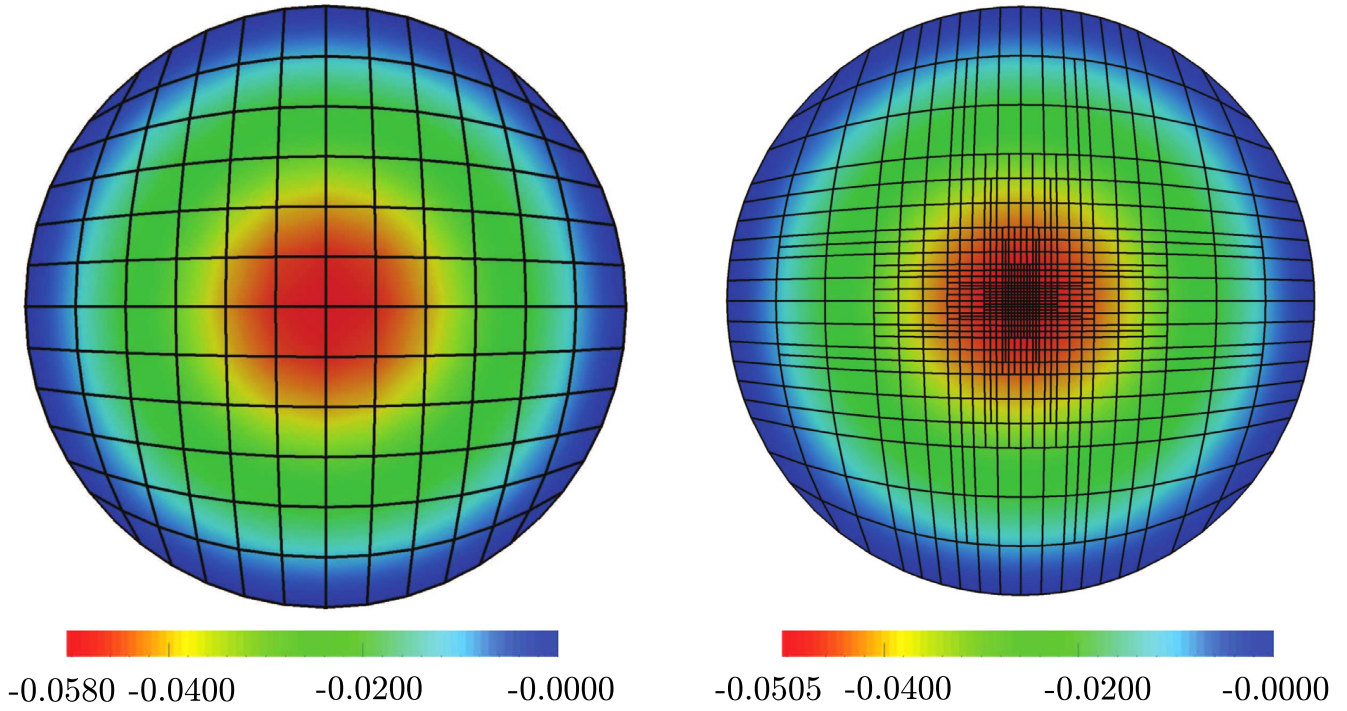


Fig. 15. (Color online) Initial NURBS mesh (left) and final quintic ASTS mesh (right) along with the corresponding computed deflection fields. The mesh on the right hand side has five levels of refinement. We used $p = 5$ in both cases.

plate's center. The equations that govern this problem can be obtained from Eqs. (12)–(16) taking $\Gamma_w = \Gamma$, $\Gamma_M = \Gamma$, $w_\Gamma = 0$, and $M_\Gamma = 0$, yielding

$$D\Delta^2 w = g \quad \text{in } \Omega, \tag{34}$$

$$w = 0 \quad \text{on } \Gamma, \tag{35}$$

$$\nu D\Delta w + (1-\nu) D\boldsymbol{n} \cdot (\nabla\nabla w)\boldsymbol{n} = 0 \quad \text{on } \Gamma. \tag{36}$$

The Poisson ratio is assumed to be $\nu = 0.3$ and the rest of the material properties are selected such that $D = 1$. Finally, g is a load per unit surface which approximates the point load P. We model g as a function of the spline space called g^h. Let us assume for the time being that our discretization produces a Greville point in the plate's center, which can be achieved, for example, using a judicious geometrical mapping. As a precursor to g^h, let us define p^h as a function living in the discrete space which takes the value one in the Greville point that falls in the center of the plate and vanishes in the remaining Greville points. Therefore, we take $p^h = \sum_{A=1}^{n} p_A R_A$ such that

$$p^h(\boldsymbol{\tau}_A) = \begin{cases} 1 & \text{if } \boldsymbol{\tau}_A \text{ falls in the plate's center,} \\ 0 & \text{otherwise.} \end{cases} \tag{37}$$

The function g^h is simply defined as $g^h = \lambda p^h$, where the constant λ is determined imposing that g^h is statically equivalent to P, that is,

$$\int_\Omega p^h \mathrm{d}\Omega = P, \tag{38}$$

which completely defines g^h.

For the considered example, it is known that the exact solution for the deflection at the center of the plate is

$$w_0^{ex} = \frac{Pr^2}{16\pi D}\left(\frac{3+\nu}{1+\nu}\right). \tag{39}$$

We solve this problem using quintic ASTS. We start with a uniform NURBS mesh and end up with five levels of refinement in order to have higher resolution close to the center of the plate. Fig. 15 shows the coarsest and finest meshes

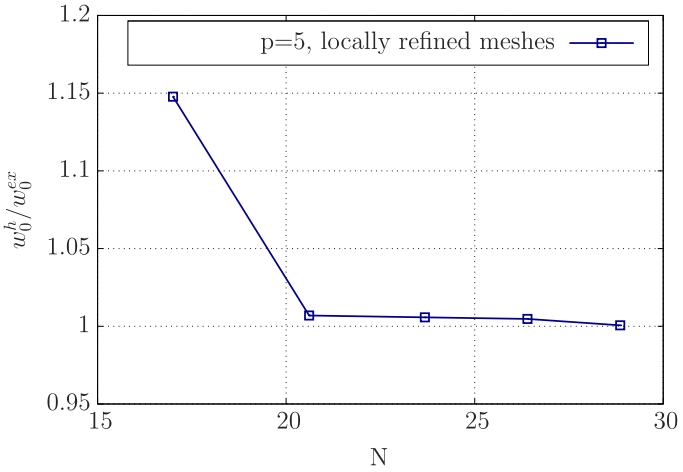


Fig. 16. (Color online) Convergence of the numerical deflection at the center of the plate using quintic ASTS. We start with a uniform NURBS mesh and introduce a level of refinement in each new mesh. N denotes the square root of the total number of degrees of freedom.

along with the corresponding computed deflection fields. In Fig. 16, we plot the convergence of the deflection of the plate's center normalized with respect to the exact solution. A good convergence behavior is observed, and the relative error of the numerical solution with respect to the exact solution for the mesh with five levels of refinement is 0.06%.

Remark 4.3. We acknowledge that our simple strategy for imposing point loads requires to have a Greville point where the load is applied. Although this can always be achieved with a judicious mapping, in the future, we will investigate in more detail the possibility of developing more general strategies to impose point loads within the isogeometric collocation framework, taking also inspiration from works like [73].

5. Conclusions

This paper presents novel locally refinable isogeometric collocation methods based on analysis-suitable T-splines of arbitrary degree. The combination of isogeometric collocation and T-splines allows local refinement while keeping one point evaluation per degree of freedom, has good approximation properties, and is geometrically flexible. Moreover, the proposed methods are stable, accurate, and robust as shown by several significant numerical examples, where both second- and fourth-order boundary-value problems have been considered. We believe that the methodologies developed herein are highly suitable to tackle in an efficient way a large variety of demanding problems for which local refinement capabilities constitute a major advantage. Among others, these include shells and contact problems, topics that we plan to study in the near future.

Acknowledgments

Part of this work was accomplished during a four-month visit of HC at Carnegie Mellon University in 2014, for which financial support from the Ph.D. student grant UDC-Inditex is gratefully acknowledged. HG and HC were partially supported by the European Research Council through the FP7 Ideas Starting Grant (project # 307201). LL and YZ were partially supported by the PECASE Award (project # N00014-14-1–0234). AR was partially supported by the European Research Council through the FP7 Ideas Starting Grant (project # 259229). This support is gratefully acknowledged. Finally, we acknowledge the open source scientific library PETSc and their developers.

Appendix. High-order spatial derivatives of rational T-spline basis functions

Our starting point is the elemental geometric mapping introduced in Eq. (3). Assuming this mapping is invertible, we define its inverse function $\boldsymbol{\xi}^e : \Omega^e \mapsto \square$ such that $\boldsymbol{x}^e(\boldsymbol{\xi}^e(\boldsymbol{x})) = \boldsymbol{x}$ for all $\boldsymbol{x} \in \Omega^e$. Taking the derivative of this expression, we obtain the relation

$$\text{(repeated subscripts sum)} \quad x^e_{i,\alpha}\xi^e_{\alpha,j} = \delta_{ij}, \tag{A.1}$$

where x_i^e is the ith component of $\boldsymbol{x}^e$, ξ_α^e is the αth component of $\boldsymbol{\xi}^e$, and δ_{ij} is the Kronecker's delta. Latin and Greek indices denote spatial and parametric coordinates, respectively. An index after a comma denotes partial differentiation, for example, $x_{i,\alpha}^e = \partial x_i^e / \partial \xi_\alpha$. To obtain the derivatives of the T-spline blending functions in physical space, we start with the following identity

$$R_A\left(\mathbf{x}^e\left(\boldsymbol{\xi}\right)\right) = R_a^e\left(\boldsymbol{\xi}\right) \quad \boldsymbol{\xi} \in \square, \tag{A.2}$$

Eq. (A.2) simply expresses the correspondence between basis functions in physical and parametric space. If we compute the derivative of Eq. (A.2), it follows that

$$\text{(repeated subscripts sum)} \quad R_{A,i} x_{i,\alpha}^e = R_{a,\alpha}^e. \tag{A.3}$$

Multiplying this equation with $\xi_{\alpha,j}^e$ and using Eq. (A.1), we can easily obtain $R_{A,i}$ as

$$\text{(repeated subscripts sum)} \quad R_{A,i} = R_{a,\alpha}^e \xi_{\alpha,i}^e. \tag{A.4}$$

The higher-order spatial derivatives are obtained analogously, that is, deriving Eq. (A.2) several times by way of the chain rule, and solving for the spatial derivatives using Eq. (A.1). Proceeding this way, we obtain the second-, the third-, and the fourth-order derivatives, which take on the expressions

$$\text{(repeated subscripts sum)} \quad R_{A,ij} = [R_{a,\alpha\beta}^e - R_{A,m} x_{m,\alpha\beta}^e] \xi_{\alpha,i}^e \xi_{\beta,j}^e, \tag{A.5}$$

$$\begin{aligned}\text{(repeated subscripts sum)} \quad R_{A,ijk} = {} & [R_{a,\alpha\beta\gamma}^e - R_{A,mn}(x_{n,\beta\gamma}^e x_{m,\alpha}^e + x_{m,\alpha\gamma}^e x_{n,\beta}^e + x_{m,\alpha\beta}^e x_{n,\gamma}^e) \\ & - R_{A,m} x_{m,\alpha\beta\gamma}^e] \xi_{\alpha,i}^e \xi_{\beta,j}^e \xi_{\gamma,k}^e,\end{aligned} \tag{A.6}$$

$$\begin{aligned}\text{(repeated subscripts sum)} \quad R_{A,ijkl} = {} & [R_{a,\alpha\beta\gamma\delta}^e - R_{A,mnq}(x_{q,\gamma\delta}^e x_{n,\beta}^e x_{m,\alpha}^e + x_{n,\beta\delta}^e x_{q,\gamma}^e x_{m,\alpha}^e \\ & + x_{m,\alpha\delta}^e x_{q,\gamma}^e x_{n,\beta}^e + x_{q,\delta}^e (x_{n,\beta\gamma}^e x_{m,\alpha}^e + x_{m,\alpha\gamma}^e x_{n,\beta}^e + x_{m,\alpha\beta}^e x_{n,\gamma}^e)) \\ & - R_{A,mn}(x_{n,\beta\gamma\delta}^e x_{m,\alpha}^e + x_{n,\beta\gamma}^e x_{m,\alpha\delta}^e + x_{n,\beta\delta}^e x_{m,\alpha\gamma}^e + x_{m,\alpha\gamma\delta}^e x_{n,\beta}^e \\ & + x_{n,\gamma\delta}^e x_{m,\alpha\beta}^e + x_{m,\alpha\beta\delta}^e x_{n,\gamma}^e + x_{m,\alpha\beta\gamma}^e x_{n,\delta}^e) \\ & - R_{A,m} x_{m,\alpha\beta\gamma\delta}^e] \xi_{\alpha,i}^e \xi_{\beta,j}^e \xi_{\gamma,k}^e \xi_{\delta,l}^e.\end{aligned} \tag{A.7}$$

The above formulas give the spatial derivatives of R_A in terms of the parametric derivatives of its local counterpart R_a^e. However, the functions R_a^e are rational splines defined as

$$R_a^e\left(\boldsymbol{\xi}\right) = \frac{w_a^e N_a^e\left(\boldsymbol{\xi}\right)}{w^e(\boldsymbol{\xi})}, \quad \text{where } w^e(\boldsymbol{\xi}) = \sum_{b=1}^{n^e} w_b^e N_b^e(\boldsymbol{\xi}). \tag{A.8}$$

The function N_a^e is the lowest-level function in an isogeometric code, so let us compute the derivatives of R_a^e in terms of those of N_a^e. Using basic manipulations, we obtain

$$R_{a,\alpha}^e = \frac{w_a^e N_{a,\alpha}^e}{w^e} - \frac{w_a^e N_a^e w_{,\alpha}^e}{(w^e)^2}, \tag{A.9}$$

which may be rewritten as

$$R_{a,\alpha}^e = \frac{w_a^e N_{a,\alpha}^e - R_a^e w_{,\alpha}^e}{w^e}. \tag{A.10}$$

Proceeding analogously, we obtain

$$R^e_{a,\alpha\beta} = \frac{w^e_a N^e_{a,\alpha\beta} - R^e_a w^e_{,\alpha\beta} - R^e_{a,\beta} w^e_{,\alpha} - R^e_{a,\alpha} w^e_{,\beta}}{w^e}, \tag{A.11}$$

$$\begin{aligned} R^e_{a,\alpha\beta\gamma} = & \frac{w^e_a N^e_{a,\alpha\beta\gamma} - R^e_a w^e_{,\alpha\beta\gamma} - R^e_{a,\alpha} w^e_{,\beta\gamma} - R^e_{a,\beta} w^e_{,\alpha\gamma}}{w^e} \\ & - \frac{R^e_{a,\gamma} w^e_{,\alpha\beta} + R^e_{a,\beta\gamma} w^e_{,\alpha} + R^e_{a,\alpha\gamma} w^e_{,\beta} + R^e_{a,\alpha\beta} w^e_{,\gamma}}{w^e}, \end{aligned} \tag{A.12}$$

$$\begin{aligned} R^e_{a,\alpha\beta\gamma\delta} = & \frac{w^e_a N^e_{a,\alpha\beta\gamma\delta} - R^e_{a,\delta} w^e_{,\alpha\beta\gamma} - R^e_a w^e_{,\alpha\beta\gamma\delta} - R^e_{a,\alpha\delta} w^e_{,\beta\gamma} - R^e_{a,\alpha} w^e_{,\beta\gamma\delta}}{w^e} \\ & - \frac{R^e_{a,\beta\delta} w^e_{,\alpha\gamma} + R^e_{a,\beta} w^e_{,\alpha\gamma\delta} + R^e_{a,\gamma\delta} w^e_{,\alpha\beta} + R^e_{a,\gamma} w^e_{,\alpha\beta\delta} + R^e_{a,\beta\gamma\delta} w^e_{,\alpha}}{w^e} \\ & - \frac{R^e_{a,\beta\gamma} w^e_{,\alpha\delta} + R^e_{a,\alpha\gamma\delta} w^e_{,\beta} + R^e_{a,\alpha\gamma} w^e_{,\beta\delta} + R^e_{a,\alpha\beta\delta} w^e_{,\gamma} + R^e_{a,\alpha\beta} w^e_{,\gamma\delta}}{w^e} - \frac{R^e_{a,\alpha\beta\gamma} w^e_{,\delta}}{w^e}. \end{aligned} \tag{A.13}$$

The above formulas completely describe the computations needed to obtain the derivatives of the basis functions in physical space up to order four.

References

[1] T.J.R. Hughes, J.A. Cottrell, Y. Bazilevs, Isogeometric analysis CAD, finite elements, NURBS, exact geometry and mesh refinement, Comput. Methods Appl. Mech. Engrg. 194 (2005) 4135–4195.

[2] J.A. Cottrell, T.J.R. Hughes, Y. Bazilevs, Isogeometric Analysis Toward Integration of CAD and FEA, Wiley, 2009.

[3] T. Elguedj, Y. Bazilevs, V.M. Calo, T.J.R. Hughes, $\overline{B}$ and $\overline{F}$ projection methods for nearly incompressible linear and non-linear elasticity and plasticity using higher-order NURBS elements, Comput. Methods Appl. Mech. Engrg. 197 (2008) 2732–2762.

[4] J. Caseiro, R. Valente, A. Reali, J. Kiendl, F. Auricchio, R. Alves de Sousa, Assumed natural strain NURBS-based solid-shell element for the analysis of large deformation elasto-plastic thin-shell structures, Comput. Methods Appl. Mech. Engrg. 284 (2015) 861–880.

[5] R. Dhote, H. Gomez, R. Melnik, J. Zu, Isogeometric analysis of a dynamic thermo-mechanical phase-field model applied to shape memory alloys, Comput. Mech. 53 (2013) 1235–1250.

[6] F. Auricchio, M. Conti, M. Ferraro, S. Morganti, A. Reali, R. Taylor, Innovative and efficient stent flexibility simulations based on isogeometric analysis, Comput. Methods Appl. Mech. Engrg. 295 (2015) 347–361.

[7] T.J.R. Hughes, J. Evans, A. Reali, Finite element and NURBS approximations of eigenvalue, boundary-value, and initial-value problems, Comput. Methods Appl. Mech. Engrg. 272 (2014) 290–320.

[8] J.A. Cottrell, A. Reali, Y. Bazilevs, T.J.R. Hughes, Isogeometric analysis of structural vibrations, Comput. Methods Appl. Mech. Engrg. 195 (2006) 5257–5296.

[9] L. Greco, M. Cuomo, B-spline interpolation of Kirchhoff-Love space rods, Comput. Methods Appl. Mech. Engrg. 256 (2013) 251–269.

[10] Y. Bazilevs, V.M. Calo, J.A. Cottrell, T.J.R. Hughes, A. Reali, G. Scovazzi, Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows, Comput. Methods Appl. Mech. Engrg. 197 (2007) 173–201.

[11] J. Liu, H. Gomez, J.A. Evans, T.J.R. Hughes, C.M. Landis, Functional entropy variables: A new methodology for deriving thermodynamically consistent algorithms for complex fluids, with particular reference to the isothermal Navier–Stokes–Korteweg equations, J. Comput. Phys. 248 (2013) 47–86.

[12] H. Gomez, T.J.R. Hughes, X. Nogueira, V.M. Calo, Isogeometric analysis of the isothermal Navier–Stokes–Korteweg equations, Comput. Methods Appl. Mech. Engrg. 199 (2010) 1828–1840.

[13] Y. Bazilevs, V.M. Calo, T.J.R. Hughes, Y. Zhang, Isogeometric fluid–structure interaction: Theory, algorithms, and computations, Comput. Mech. 43 (2008) 3–37.

[14] Y. Bazilevs, M.-C. Hsu, J. Kiendl, R. Wuchner, K.-U. Bletzinger, 3D simulation of wind turbine rotors at full scale. Part II: Fluid-structure interaction modeling with composite blades, Internat. J. Numer. Methods Fluids 65 (2011) 236–253.

[15] H. Casquero, C. Bona-Casas, H. Gomez, A NURBS-based immersed methodology for fluid–structure interaction, Comput. Methods Appl. Mech. Engrg. 284 (2015) 943–970.

[16] D. Kamensky, M.-C. Hsu, D. Schillinger, J.A. Evans, A. Aggarwal, Y. Bazilevs, M.S. Sacks, T.J.R. Hughes, An immersogeometric variational framework for fluid–structure interaction: Application to bioprosthetic heart valves, Comput. Methods Appl. Mech. Engrg. 284 (2015) 1005–1053.

[17] J. Bueno, C. Bona-Casas, Y. Bazilevs, H. Gomez, Interaction of complex fluids and solids: theory, algorithms and application to phase-change-driven implosion, Comput. Mech. 55 (6) (2015) 1105–1118.

[18] H. Gomez, V.M. Calo, Y. Bazilevs, T.J.R. Hughes, Isogeometric analysis of the Cahn-Hilliard phase-field model, Comput. Methods Appl. Mech. Engrg. 197 (49–50) (2008) 4333–4352.

[19] H. Gomez, L. Cueto-Felgueroso, R. Juanes, Three-dimensional simulation of unstable gravity-driven infiltration of water into a porous medium, J. Comput. Phys. 238 (2013) 217–239.

[20] G. Vilanova, I. Colominas, H. Gomez, Coupling of discrete random walks and continuous modeling for three-dimensional tumor-induced angiogenesis, Comput. Mech. 53 (2013) 449–464.
[21] J. Kiendl, K.-U. Bletzinger, J. Linhard, R. Wuchner, Isogeometric shell analysis with Kirchhoff-Love elements, Comput. Methods Appl. Mech. Engrg. 198 (2009) 3902–3914.
[22] J. Kiendl, M.-C. Hsu, M.C. Wu, A. Reali, Isogeometric Kirchhoff–Love shell formulations for general hyperelastic materials, Comput. Methods Appl. Mech. Engrg. 291 (2015) 280–303.
[23] C. Adam, T.J.R. Hughes, S. Bouabdallah, M. Zarroug, H. Maitournam, Selective and reduced numerical integrations for NURBS-based isogeometric analysis, Comput. Methods Appl. Mech. Engrg. 284 (2015) 732–761.
[24] F. Auricchio, L. Beirão Da Veiga, T.J.R. Hughes, A. Reali, G. Sangalli, Isogeometric collocation methods, Math. Models Methods Appl. Sci. 20 (2010) 2075–2107.
[25] F. Auricchio, L. Bẽirao da Veiga, T.J.R. Hughes, A. Reali, G. Sangalli, Isogeometric collocation for elastostatics and explicit dynamics, Comput. Methods Appl. Mech. Engrg. 249 (2012) 2–14.
[26] D. Schillinger, J. Evans, A. Reali, M. Scott, T.J.R. Hughes, Isogeometric collocation: Cost comparison with Galerkin methods and extension to adaptive hierarchical NURBS discretizations, Comput. Methods Appl. Mech. Engrg. 267 (2013) 170–232.
[27] H. Gomez, A. Reali, G. Sangalli, Accurate, efficient, and (iso)geometrically flexible collocation methods for phase-field models, J. Comput. Phys. 262 (2014) 153–171.
[28] L. De Lorenzis, J. Evans, T.J.R. Hughes, A. Reali, Isogeometric collocation: Neumann boundary conditions and contact, Comput. Methods Appl. Mech. Engrg. 284 (2015) 21–54.
[29] R. Kruse, N. Nguyen-Thanh, L. De Lorenzis, T.J.R. Hughes, Isogeometric collocation for large deformation elasticity and frictional contact problems, Comput. Methods Appl. Mech. Engrg. 296 (2015) 73–112.
[30] A. Reali, H. Gomez, An isogeometric collocation approach for Bernoulli-Euler beams and Kirchhoff plates, Comput. Methods Appl. Mech. Engrg. 284 (2015) 623–636.
[31] L. Beirão da Veiga, C. Lovadina, A. Reali, Avoiding shear locking for the Timoshenko beam problem via isogeometric collocation methods, Comput. Methods Appl. Mech. Engrg. 241–244 (2012) 38–51.
[32] F. Auricchio, L. Beirão da Veiga, J. Kiendl, C. Lovadina, A. Reali, Locking-free isogeometric collocation methods for spatial Timoshenko rods, Comput. Methods Appl. Mech. Engrg. 263 (2013) 113–126.
[33] J. Kiendl, F. Auricchio, L. Beirão da Veiga, C. Lovadina, A. Reali, Isogeometric collocation methods for the Reissner–Mindlin plate problem, Comput. Methods Appl. Mech. Engrg. 284 (2015) 489–507.
[34] J. Kiendl, F. Auricchio, T.J.R. Hughes, A. Reali, Single-variable formulations and isogeometric discretizations for shear deformable beams, Comput. Methods Appl. Mech. Engrg. 284 (2015) 988–1004.
[35] C. Manni, A. Reali, H. Speleers, Isogeometric collocation methods with generalized B-splines, Comput. Math. Appl. 70 (7) (2015) 1659–1675.
[36] T.W. Sederberg, J. Zheng, A. Bakenov, A. Nasri, T-splines and T-NURCCs, ACM Trans. Graph. 22 (2003) 477–484.
[37] T.W. Sederberg, J. Zheng, X. Song, Knot intervals and multi-degree splines, Comput. Aided Geom. Design 20 (7) (2003) 455–468.
[38] T.W. Sederberg, D.L. Cardon, G.T. Finnigan, N.S. North, J. Zheng, T. Lyche, T-spline simplification and local refinement, in: ACM Transactions on Graphics, TOG, vol. 23, ACM, 2004, pp. 276–283.
[39] Y. Bazilevs, V. Calo, J. Cottrell, J. Evans, T.J.R. Hughes, S. Lipton, M. Scott, T. Sederberg, Isogeometric analysis using T-splines, Comput. Methods Appl. Mech. Engrg. 199 (2010) 229–263.
[40] X. Li, J. Zheng, T.W. Sederberg, T.J.R. Hughes, M.A. Scott, On linear independence of T-spline blending functions, Comput. Aided Geom. Design 29 (2012) 63–76.
[41] L. Beirão da Veiga, A. Buffa, G. Sangalli, R. Vázquez, Analysis suitable T-splines of arbitrary degree: Definition, linear independence, and approximation properties, Math. Models Methods Appl. Sci. 23 (11) (2013) 1979–2003.
[42] L. Beirão da Veiga, A. Buffa, D. Cho, G. Sangalli, Analysis-suitable T-splines are dual-compatible, Comput. Methods Appl. Mech. Engrg. 249 (2012) 42–51.
[43] M. Scott, X. Li, T. Sederberg, T.J.R. Hughes, Local refinement of analysis-suitable T-splines, Comput. Methods Appl. Mech. Engrg. 213–216 (2012) 206–222.
[44] Y. Zhang, W. Wang, T.J.R. Hughes, Solid T-spline construction from boundary representations for genus-zero geometry, Comput. Methods Appl. Mech. Engrg. 249–252 (2012) 185–197.
[45] Y. Zhang, W. Wang, T.J.R. Hughes, Conformal solid T-spline construction from boundary T-spline representations, Comput. Mech. 51 (2013) 1051–1059.
[46] W. Wang, Y. Zhang, L. Liu, T.J.R. Hughes, Trivariate solid T-spline construction from boundary triangulations with arbitrary genus topology, Comput.-Aided Des. 45 (2) (2013) 351–360.
[47] L. Liu, Y. Zhang, T.J.R. Hughes, M.A. Scott, T.W. Sederberg, Volumetric T-spline construction using boolean operations, Eng. Comput. 30 (2014) 425–439.
[48] L. Liu, Y. Zhang, Y. Liu, W. Wang, Feature-preserving T-mesh construction using skeleton-based polycubes, Comput.-Aided Des. 58 (2015) 162–172.
[49] L. Liu, Y. Zhang, X. Wei, Weighted T-splines with application in reparameterizing trimmed NURBS surfaces, Comput. Methods Appl. Mech. Engrg. 295 (2015) 108–126.
[50] W. Wang, Y. Zhang, M.A. Scott, T.J.R. Hughes, Converting an unstructured quadrilateral mesh to a standard T-spline surface, Comput. Mech. 48 (2011) 477–498.
[51] M. Borden, C. Verhoosel, M. Scott, T.J.R. Hughes, C. Landis, A phase-field description of dynamic brittle fracture, Comput. Methods Appl. Mech. Engrg. 217–220 (2012) 77–95.
[52] Y. Bazilevs, M.-C. Hsu, M. Scott, Isogeometric fluid–structure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines, Comput. Methods Appl. Mech. Engrg. 249–252 (2012) 28–41.

[53] H. Casquero, L. Lei, C. Bona-Casas, J. Zhang, H. Gomez, A hybrid variational-collocation immersed method for fluid–structure interaction using unstructured T-splines, Internat. J. Numer. Methods Engrg. (2015) http://dx.doi.org/10.1002/nme.5004.
[54] C.V. Verhoosel, M.A. Scott, R. de Borst, T.J.R. Hughes, An isogeometric approach to cohesive zone modeling, Internat. J. Numer. Methods Engrg. 87 (2011) 336–360.
[55] C.V. Verhoosel, M.A. Scott, T.J.R. Hughes, R. de Borst, An isogeometric analysis approach to gradient damage models, Internat. J. Numer. Methods Engrg. 86 (2011) 115–134.
[56] R. Dimitri, L.D. Lorenzis, M. Scott, P. Wriggers, R. Taylor, G. Zavarise, Isogeometric large deformation frictionless contact using T-splines, Comput. Methods Appl. Mech. Engrg. 269 (2014) 394–414.
[57] R. Simpson, M. Scott, M. Taus, D. Thomas, H. Lian, Acoustic isogeometric boundary element analysis, Comput. Methods Appl. Mech. Engrg. 269 (2014) 265–290.
[58] A. Buffa, G. Sangalli, R. Vázquez, Isogeometric methods for computational electromagnetics: B-spline and T-spline discretizations, J. Comput. Phys. 257 (B) (2014) 1291–1320.
[59] M.-C. Hsu, D. Kamensky, F. Xu, J. Kiendl, C. Wang, M. Wu, J. Mineroff, A. Reali, Y. Bazilevs, M. Sacks, Dynamic and fluid–structure interaction simulations of bioprosthetic heart valves using parametric design with T-splines and Fung-type material models, Comput. Mech. 55 (2015) 1211–1225.
[60] H. Casquero, L. Lei, Y. Zhang, A. Reali, J. Kiendl, H. Gomez, Arbitrary-degree T-splines for isogeometric analysis of fully nonlinear Kirchhoff-Love shells, submitted for publication.
[61] G. Finnigan, Arbitrary degree T-splines (Masters thesis), Brigham Young University, 2008.
[62] M.J. Borden, M.A. Scott, J.A. Evans, T.J.R. Hughes, Isogeometric finite element data structures based on Bézier extraction of NURBS, Internat. J. Numer. Methods Engrg. 87 (1–5) (2011) 15–47.
[63] M.A. Scott, M.J. Borden, C.V. Verhoosel, T.W. Sederberg, T.J.R. Hughes, Isogeometric finite element data structures based on Bézier extraction of T-splines, Internat. J. Numer. Methods Engrg. 88 (2011) 126–156.
[64] L. Lei, H. Casquero, H. Gomez, J. Zhang, Hybrid-degree weighted T-splines and their application in isogeometric analysis, submitted for publication.
[65] M. Scott, R. Simpson, J. Evans, S. Lipton, S. Bordas, T. Hughes, T. Sederberg, Isogeometric boundary element analysis using unstructured T-splines, Comput. Methods Appl. Mech. Engrg. 254 (2013) 197–221.
[66] L. Liu, Y. Zhang, X. Wei, Handling extraordinary nodes with weighted T-spline basis functions, in: 24th International Meshing Roundtable, 2015, in press.
[67] T.J.R. Hughes, The Finite Element Method: Linear Static and Dynamic Finite Element Analysis.
[68] S. Balay, M.F. Adams, J. Brown, P. Brune, K. Buschelman, V. Eijkhout, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, K. Rupp, B.F. Smith, H. Zhang, PETSc Web page, http://www.mcs.anl.gov/petsc, 2014.
[69] S. Balay, M.F. Adams, J. Brown, P. Brune, K. Buschelman, V. Eijkhout, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, K. Rupp, B.F. Smith, H. Zhang, PETSc Users Manual, Tech. Rep. ANL-95/11 - Revision 3.4, Argonne National Laboratory, 2013.
[70] A. Brooks, T.J.R. Hughes, Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations, Comput. Methods Appl. Mech. Engrg. 32 (1982) 199–259.
[71] L.P. Franca, S.L. Frey, T.J. Hughes, Stabilized finite element methods: I. Application to the advective-diffusive model, Comput. Methods Appl. Mech. Engrg. 95 (2) (1992) 253–276.
[72] H. Gomez, I. Colominas, F. Navarrina, J. París, M. Casteleiro, A hyperbolic theory for advection-diffusion problems: Mathematical foundations and numerical modeling, Arch. Comput. Methods Eng. 17 (2) (2010) 191–211.
[73] Q. Dang, M. Ehrhardt, On Dirac delta sequences and their generating functions, Appl. Math. Lett. 25 (2012) 2385–2390.

C Paper #3: "A hybrid variational-collocation immersed method for fluid-structure interaction using unstructured T-splines"

INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2016; **105**:855–880
Published online 8 September 2015 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/nme.5004

A hybrid variational-collocation immersed method for fluid-structure interaction using unstructured T-splines

Hugo Casquero[1,*,†], Lei Liu[2], Carles Bona-Casas[1], Yongjie Zhang[2] and Hector Gomez[1]

[1]*Departamento de Métodos Matemáticos, Universidade da Coruña, Campus de A Coruña, 15071, A Coruña, Spain*
[2]*Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA*

SUMMARY

We present a hybrid variational-collocation, immersed, and fully-implicit formulation for fluid-structure interaction (FSI) using unstructured T-splines. In our immersed methodology, we define an Eulerian mesh on the whole computational domain and a Lagrangian mesh on the solid domain, which moves arbitrarily on top of the Eulerian mesh. Mathematically, the problem reduces to solving three equations, namely, the linear momentum balance, mass conservation, and a condition of kinematic compatibility between the Lagrangian displacement and the Eulerian velocity. We use a weighted residual approach for the linear momentum and mass conservation equations, but we discretize directly the strong form of the kinematic relation, deriving a hybrid variational-collocation method. We use T-splines for both the spatial discretization and the information transfer between the Eulerian mesh and the Lagrangian mesh. T-splines offer us two main advantages against non-uniform rational B-splines: they can be locally refined and they are unstructured. The generalized-α method is used for the time discretization. We validate our formulation with a common FSI benchmark problem achieving excellent agreement with the theoretical solution. An example involving a partially immersed solid is also solved. The numerical examples show how the use of T-junctions and extraordinary nodes results in an accurate, efficient, and flexible method.

Received 29 April 2015; Revised 31 July 2015; Accepted 31 July 2015

KEY WORDS: fluid-structure interaction; isogeometric analysis; mmersed methods; analysis-suitable T-splines; variational multiscale; collocation

1. INTRODUCTION

Isogeometric analysis (IGA) was presented by Hughes *et al.* in [1] and explained in detail in [2]. IGA was invented to fill the gap between computer aided design (CAD) and finite element analysis (FEA) in order to reduce the communication process between design and analysis. IGA's key idea is to utilize the blending functions of CAD in FEA to represent both the geometry and the solution using the isoparametric concept. Different CAD functions may be used, including non-uniform rational B-Splines (NURBS), T-splines, and subdivision surfaces. NURBS-based IGA is completely widespread nowadays. An important aftereffect of using NURBS as a basis in analysis is that we can take advantage of their natural inter-element smoothness, which has positive consequences in several application areas, such as, body-fitted fluid-structure interaction (FSI) [3–6], immersed FSI [7–9], fluid mechanics [10–15], phase-field models [16–20], biomechanics [21, 22], structural mechanics [23–26], shape memory alloys [27–29], shell modeling [30–33], and contact problems [34–36], among others.

*Correspondence to: Hugo Casquero, Departamento de Métodos Matemáticos, Universidade da Coruña, Campus de A Coruña, 15071, A Coruña, Spain.
†E-mail: hugo.casquero@udc.es

1.1. Isogeometric analysis: from NURBS to T-splines

Non-uniform rational B-splines were the CAD technology of choice when isogeometric analysis was proposed. NURBS are widely used in engineering design and have a number of appealing features, such as, for example, higher-order global continuity, the convex hull property, the ability to represent exactly all quadric curves and surfaces, affine covariance, pointwise non-negativity, the variation diminishing property, and capability for free-form surface modeling. Nevertheless, NURBS have some major limitations such as the difficulty to represent watertight surfaces. Another shortcoming is their tensor product structure, which avoids local refinement or unstructured meshing. In [37], Sederberg *et al.* introduced T-spline surfaces in CAD as a generalization of NURBS. T-splines overcome the aforementioned limitations of NURBS while maintaining all their good properties. In the last few years, Zhang *et al.* proposed several procedures to create volumetric T-splines [38–42]. T-splines are a superset of NURBS. An appealing feature of T-splines is that they are forward and backward compatible with NURBS. In particular, any T-spline surface can be converted into one or more NURBS surfaces by performing repeated local refinement to eliminate all T-junctions. In the commercial field, this is a significant advantage of T-splines over other candidates that are unstructured and allow local refinement (such as subdivision surfaces) but are not compatible with NURBS.

In [43], T-splines were introduced in the FEA field. In contrast with NURBS, T-spline blending functions as introduced in [37] do not define a basis because they might be linearly dependent. Linear independence is not necessary in CAD, but it is a requirement for successful analysis. This problem was addressed defining a subset of T-splines where linear independence is guaranteed [44, 45]. The newly proposed subset of T-splines was called analysis-suitable T-splines (ASTS). Also, to facilitate the direct inclusion of T-splines in a finite element code, the concept of Bézier extraction was proposed [46, 47]. Significant developments have occurred since then: the local refinement of ASTS was analyzed in [48], the conversion of an unstructured quadrilateral mesh to a standard T-spline surface was detailed in [49], the conversion of a trimmed NURBS surface to an untrimmed T-spline surface was explained in [50], the application of the hierarchical refinement idea to ASTS was explored in [51], and the application of collocation methods [52–55] to ASTS was studied in [56]. ASTS have been applied to several problems in computational mechanics such as dynamic brittle fracture [57], cohesive zone modeling [58], gradient damage models [59], large deformation frictionless contact [60], and acoustics [61]. However, their use in FSI, which is precisely the goal of this paper, has been limited to a body-fitted method where a bicubic T-spline surface discretization of a rotation-free Kirchhoff–Love shell model was used for the solid and volumetric quadratic NURBS were used for the fluid [62].

1.2. Fluid-structure interaction methods

Fluid-structure interaction methods are typically divided into interface-tracking and interface-capturing techniques. In interface-tracking methods, as the fluid domain changes its shape due to the fluid–solid interface motion, the fluid mesh is updated to follow this movement. In interface-capturing methods, the fluid mesh does not move to track the fluid–solid interface. Interface-tracking (i.e., moving-mesh) methods are often based on the arbitrary Lagrangian–Eulerian (ALE) formulation [63]. The ALE description is employed in the fluid domain, whereas the Lagrangian description is used in the solid. ALE-based FSI methods have been used in various applications, such as, for example, hemodynamics [5, 64–68], wind turbines [69, 70], or aeroelasticity [71]. As an alternative to ALE-based methods, space-time body-fitted methods are becoming increasingly popular [72, 73]. For a thorough discussion about body-fitted approaches, the reader is referred to [74].

An important example of the interface-capturing approach is the immersed boundary method [75], which was widely used in the simulation of biomechanical problems [76–81]. In this immersed technique, the solid is discretized using a Lagrangian mesh[‡] that can move freely on top of a background Eulerian mesh that spans the whole computational domain. An information transfer

[‡]By Lagrangian mesh, we refer to a mesh whose nodes are attached to material (rather than physical) points. It does not imply in any way the use of classical Lagrangian finite elements, which are not utilized in this work.

DOI: 10.1002/nme

algorithm is required to couple these two independent meshes, and this was initially accomplished by way of smoothed-out Dirac delta functions. In many cases of practical interest, the immersed approach avoids the use of mesh-updating or remeshing procedures, which are needed in body-fitted FSI methods. This is an advantage of the immersed method because changing the mesh, either geometrically or topologically, increases the computational cost and may introduce some issues. For example, when remeshing is not used for cases involving large translations or rotations of the solid, the fluid mesh will become highly distorted, compromising the accuracy of the solution. When remeshing is used, projections between different meshes are necessary; this also introduces additional inaccuracies. Thus, for FSI applications which involve large deformations or topological changes of the fluid domain, body-fitted methods may not be the best option, and this is precisely the area where immersed algorithms shine. In the 2000s, an important development occurred in the field of immersed FSI methods, namely, the introduction of the so-called immersed finite element method (IFEM) [82]. The IFEM has been successfully applied to a number of problems [83, 84].

Notably, recent research endeavors have targeted the creation of new FSI immersed methodologies using the IGA framework. This new developing type of computational techniques was termed *immersogeometric analysis*. It aims to capture the solid geometry exactly and embed it in a background mesh. This concept was proposed in [7, 9], where the authors used thin-shell structures modeled geometrically as surfaces embedded in a fluid. In those articles, various benchmark problems are considered. The results were compared with body-fitted methods, and the algorithm was applied to a cardiovascular problem involving tri-leaflet bioprosthetic heart valves. In [8], we proposed a new method which falls in the same category. In contrast with that of [7, 9], our method considers volumetric solids. The algorithm was applied to 2D and 3D benchmark problems with theoretical solution achieving excellent agreement. In addition, we showed that the higher-order continuity of IGA basis functions increases the robustness of immersed methodologies in problems involving large deformations of the solid.

We feel that IGA-based, immersed FSI methods would benefit from the use of unstructured meshes and local refinement and this motivates our current work. Here, we bring the T-spline technology to the immersed FSI realm, discretizing both the fluid and the solid with analysis-suitable T-splines. We show that the local refinement capability of T-splines may be used to achieve enhanced spatial resolution where it is needed and the unstructured nature of T-splines may help avoid the presence of singular points in the parameterization, leading to more flexible immersed FSI formulations. Additionally, we use a hybrid variational-collocation approach, namely, we discretize the linear momentum balance and mass conservation equation using the variational multiscale method and the kinematic relation between the Eulerian velocity and the Lagrangian displacement using collocation. Our hybrid variational-collocation approach enhances the efficiency of the algorithm, particularly when higher orders are used.

1.3. Structure and content of the paper

This paper is organized as follows. In Section 2, we summarize the T-spline paraphernalia. Section 3 briefly describes the immersed FSI method. We start by defining important nomenclature and describing the kinematics involved in the FSI methodology. We continue stating the main three equations involved in our methodology, which are the linear momentum balance equation, the mass conservation equation, and the kinematic relation between the Lagrangian and the Eulerian motions. Next, we discretize in space using a hybrid variational-collocation method based on T-splines. Time discretization is performed using a fully-implicit, second-order accurate algorithm based on the generalized-α method [85, 86]. Section 4 displays two numerical examples that show the performance of the proposed method. The first example is a common FSI benchmark problem, namely, a free-falling cylinder embedded in a channel. Our numerical result is checked against the theoretical solution, showing good quantitative agreement. The second example involves a deformable solid partially immersed in a flow. The solid has its bottom part fixed, what modifies the flow patterns creating a recirculation region downstream the solid. Finally, in Section 5, some concluding remarks will be drawn.

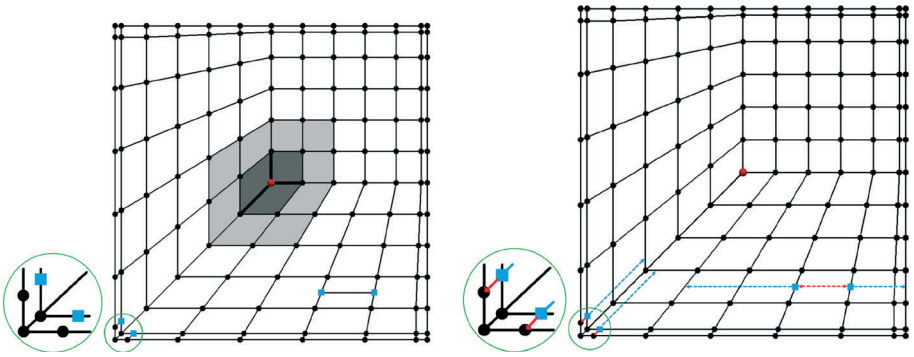
DOI: 10.1002/nme

2. T-SPLINE OVERVIEW

In this section, we present a brief introduction to the T-spline technology and remit the reader to other references for further details when it is needed. We assume that the reader is familiarized with NURBS and use them as a starting point to describe T-splines. Although we will limit ourselves to cubic T-spline surfaces, in some cases we will use the notation d_p for the number of parametric dimensions, d for the number of spatial dimensions, and p for the polynomial degree.

2.1. T-mesh, T-junctions, extraordinary nodes, and T-junction extensions

Here, we follow [45] and define the *T-mesh* as a polygonal mesh which encodes all the topological information associated to the T-spline technology. The polygons that form the T-mesh will be referred to as *faces*. The word element is used in the literature as a synonym of face, but we will reserve it for something else. We define the T-mesh in such a way that its topology is identical to that of the control mesh. Figure 1(a) shows an example of a T-mesh similar to those used in the computations presented in Section 4. Small circles and squares identify the *vertices* of the T-mesh, while a line joining (exactly) two vertices defines an *edge*. Vertices will also be referred to as *nodes*. We call *valence* the number of edges that touch a vertex. Note that our T-mesh is allowed to have T-junctions and extraordinary nodes. L-junctions and I-junctions are not allowed [44]. T-junctions are analogous to hanging nodes in classical finite elements (see the blue squares in Figure 1(a)). *Extraordinary nodes* are those interior vertices which not having valence four, do not qualify as T-junctions (see the red circular vertex in Figure 1(a)). The presence of extraordinary nodes makes the T-mesh unstructured. As we will see later, in cases of practical relevance, T-junctions will be located far away from extraordinary nodes. Therefore, in the vicinity of T-junctions, we can define a rectangular topology and a local index space [87, 88], which may be used to describe T-junction extensions exactly as in [44, 87]. Without going into the details of the definitions of the extensions, which can be found in the aforementioned papers, we just note that there are two types, namely, face extensions and edge extensions. We will also use the concept of first-bay face extension (used synonymously to one-bay face extension) as defined in [45, 88]. The *extendend T-mesh* is simply the T-mesh augmented with the T-junction extensions. As an illustration, we present in Figure 1(b) the extended T-mesh associated to the T-mesh in Figure 1(a). The blue dashed lines represent face extensions, while the red dashed lines denote edge extensions. For future reference, we also define the one-ring, two-ring and n-ring neighborhood of a T-mesh vertex. The *one-ring neighborhood* of a vertex is the set of T-mesh faces that touch that particular vertex. The *two-ring neighborhood* of a



(a) Unstructured T-mesh (b) Extended unstructured T-mesh

Figure 1. (a) Unstructured T-mesh with one extraordinary node and four T-junctions. The extraordinary node is marked with a red circle and the T-junctions are highlighted with blue squares. The spoke edges of the extraordinary node are indicated by thick black lines. The one-ring neighborhood of the extraordinary node is drawn with darkly shaded faces, while the two-ring neighborhood is the one-ring and the lightly shaded faces. (b) Extended T-mesh. The extended T-mesh is obtained by adding the T-junction extensions to the T-mesh. Face extensions are denoted by blue dashed arrows and edge extensions are denoted by red dashed arrows. The inset represents a zoom in of the bottom left corner.

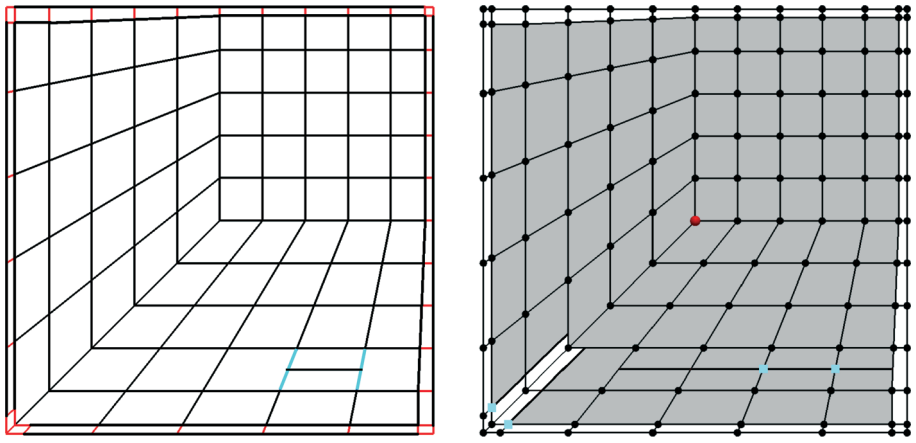
DOI: 10.1002/nme

vertex is the set of faces that comprise its one-ring neighborhood and those that touch the one-ring neighborhood. The *n-ring neighborhood* may be defined recursively in the obvious way. Figure 1(a) shows the one-ring neighborhood of the extraordinary node highlighted with darkly-shaded faces. The two-ring neighborhood is composed by the darkly-shaded and the lightly-shaded faces. Finally, we call *spoke edges* the T-mesh edges that end in an extraordinary node (see in Figure 1(a) the edges plotted with thick lines in the neighborhood of the extraordinary node).

2.2. *Knot intervals and elemental T-mesh*

We note that all the definitions given so far are purely topological and do not depend upon any geometrical information. However, to construct T-spline blending functions and perform analysis, we need to utilize geometrical information. The first step is to assign a non-negative real number to each T-mesh edge. These non-negative real numbers will be referred to as *knot intervals* [37], and they will be used to define the support of T-spline blending functions. There is significant freedom to select knot intervals, but they have to satisfy some conditions in order to define a valid knot interval configuration. A valid knot interval configuration requires that the knot intervals assigned to opposite edges of a face add up to the same value. Due to our choice to have a T-mesh with the same topology as the control mesh, we also define an outer ring of zero-length knot intervals in the T-mesh. This choice is analogous to that of using open-knot vectors in NURBS-based analysis, which leads to repeated knots on the boundary and eases the imposition of Dirichlet boundary conditions. Figure 2(a) shows a possible valid knot interval configuration associated to the T-mesh depicted in Figure 1(a). The knot intervals assigned to red, blue, and black edges take the value 0, 1/2, and 1, respectively. Note that edges associated to zero-length knot intervals are plotted with a finite length to make simpler the interpretation of the figure.

Another important object in the T-spline technology is the *elemental T-mesh*. The elemental T-mesh is a new set of polygons which might be obtained from the extended T-mesh and the knot interval configuration. The polygons that form the elemental T-mesh will be called *elements*. T-mesh faces of zero surface (i.e., zero measure) do not contribute elements to the elemental T-mesh and all T-mesh faces of non-zero surface give rise to at least one element in the elemental T-mesh. Those which are not crossed by a T-junction extension give rise to exactly one element, and those crossed by T-junction extensions may give rise to more than one. Figure 2(b) shows as shaded areas delimited by black lines the polygons that compose the elemental T-mesh. See, in particular, how the T-mesh faces in the bottom right area have been split into two elements in the elemental T-mesh. Note also how the zero-surface faces do not form elements (see the outer ring and the area in the



(a) Knot interval configuration (b) Elemental T-mesh

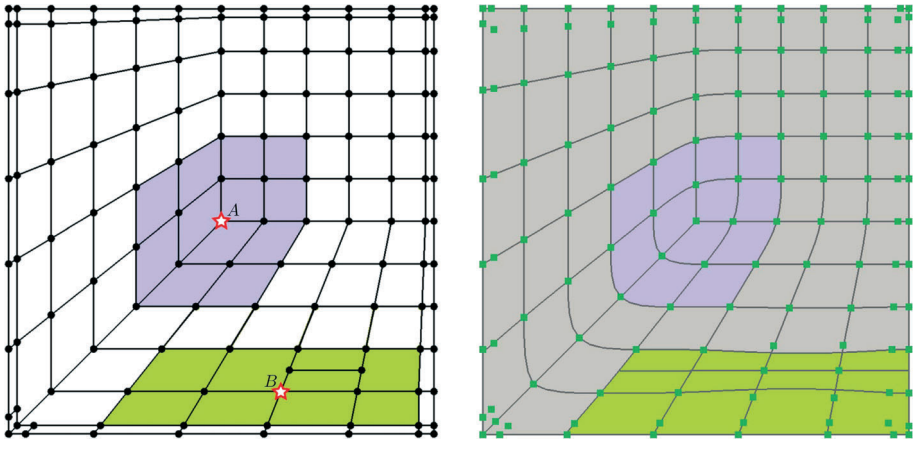
Figure 2. (a) Valid knot interval configuration associated to a particular T-mesh. The black, blue, and red edges have associated knot intervals of 1, 1/2, and 0, respectively. (b) Elemental T-mesh. The polygons which are not shaded are not part of the elemental T-mesh. T-junction extensions become new edges which may split faces into several elements of the elemental T-mesh.

DOI: 10.1002/nme

bottom left corner). The elements of the elemental T-mesh are important objects for analysis because they delimit areas in which all T-spline blending functions are $\mathcal{C}^\infty$. In analysis, the elements of the elemental T-mesh will be pushed forward to physical space using the isoparametric concept [89]. Each element of the elemental T-mesh gives rise to a so-called Bézier element in physical space. Bézier elements are suitable regions to perform numerical integration because they are bounded by lines across which at least one T-spline blending function fails to be $\mathcal{C}^\infty$. Within a Bézier element, however, all T-spline blending functions are $\mathcal{C}^\infty$.

2.3. T-spline blending functions and Bézier extraction

As indicated before, we defined the T-mesh in such a way that it is topologically identical to the control mesh. Thus, each vertex of the T-mesh has a control point associated and, therefore, a T-spline blending function. To each vertex, we associate a capital index $A \in \{1, \dots, n\}$, where n is the global number of control points. We proceed now to define the T-spline blending function associated to the vertex A, which will be denoted by N_A. Let us begin by defining the support of N_A, which can be inferred from the T-mesh and the knot interval configuration. To each vertex A of the T-mesh, we will associate a set of T-mesh faces denoted by $\mathcal{S}_A^T$. Some of the faces in $\mathcal{S}_A^T$ will have a correspondence in the elemental T-mesh, giving rise to a new set that we call $\mathcal{S}_A^{T_e}$. The region occupied by the elements in $\mathcal{S}_A^{T_e}$ will be mapped to the physical space producing a set of Bézier elements. The region defined by those Bézier elements is the support of N_A in physical space. Let us describe now $\mathcal{S}_A^T$ for an arbitrary vertex A. If A is an extraordinary node or falls in the one-ring of an extraordinary node, then $\mathcal{S}_A^T$ is the two-ring neighborhood of A. Otherwise, $\mathcal{S}_A^T$ is constructed by marching through the T-mesh in each topological direction, starting at A, until $p-1$ vertices or perpendicular edges are intersected or until the boundary of the T-mesh is encountered. Figure 3(a) shows the sets $\mathcal{S}_A^T$ and $\mathcal{S}_B^T$ for vertices A and B (marked with stars in the plot). $\mathcal{S}_A^T$ is composed of the violet faces, while $\mathcal{S}_B^T$ is comprised by the green faces. Note that the faces of zero surface in $\mathcal{S}_B^T$ have been disregarded, as they will not contribute Bézier elements in physical space [recall the knot interval configuration shown in Figure 2(a). With the aforementioned information at hand, it is possible to anticipate which basis functions will have support on a given Bézier element e, which occupies a region Ω^e in physical space. To simplify notation, we introduce a local numbering for the basis functions as it is typically carried out in the finite element method. Here, we follow [90], and use the array IEN to establish a correspondence between local and global numbering. In particular, we use the formula $A = \text{IEN}(a, e)$, where A is a global basis function index, a is a



(a) Sets $\mathcal{S}_A^T$ and $\mathcal{S}_B^T$ (b) Bézier mesh and Greville points

Figure 3. (a) Sets $\mathcal{S}_A^T$ (violet) and $\mathcal{S}_B^T$ (green) associated to vertices A and B of the T-mesh. For simplicity, we have omitted faces of zero surface in $\mathcal{S}_B^T$ as they will not contribute elements in the Bézier mesh. (b) Bézier mesh in physical space and support of the basis functions associated to vertices A (violet) and B (green). Dark green squares denote Greville points in physical space.

DOI: 10.1002/nme

local-to-element basis function counter, and e denotes the element number. Let us define the vector $M^e = \{M_a^e\}_{a=1}^{n^e}$, where the functions M_a^e are progenitors of the basis functions with support on the element e. Note that, as emphasized with the superscript e in n^e, the length of the vector M^e changes from one element to another. This is a peculiar feature of T-splines analysis which does not hold true for NURBS-based analysis.

The concept of Bézier extraction permits computing M^e as a linear combination of the canonical tensor product Bernstein polynomials defined on a fixed parent element, namely, $\square = [-1, +1]^2$. This can be expressed mathematically as

$$M^e(\xi) = C^e B(\xi), \quad \xi \in \square, \tag{1}$$

where $B = \{B_i^p\}_{i=1}^{(p+1)^{d_p}}$ is a vector containing the two-dimensional Bernstein polynomials of degree p in the domain $\square$ (see [45] for a precise description). The linear operator C^e can be represented by a rectangular matrix and it is called element Bézier extraction operator. Although we will not go into the details of the computation of C^e, which may be found in [46] and [45], we point out that the Bézier extraction operator is always computed from the knot interval configuration of the T-mesh. However, we need to distinguish two different types of elements. The elements in the two-ring of an extraordinary node are called irregular elements, while the remaining elements are called regular elements. The Bézier extraction procedure for the regular elements is based on knot insertion and was precisely described in [46]. However, the Bézier extraction procedure for the irregular elements is based on imposing basic analysis-suitability properties and geometric constraints and it was described in detail in [45]. In any case, Equation (1) applies and can be used to compute M^e as a linear combination of Bernstein polynomials. Furthermore, in practical cases, it may be useful to utilize rational functions, typically to reproduce exactly a particular geometry. Given a set of weights $\{w_A\}_{A=1}^n$ and their local counterparts for element e, namely, $\{w_a^e\}$, we can rationalize the functions in M^e as

$$N_a^e(\xi) = \frac{w_a^e M_a^e(\xi)}{\sum_{b=1}^{n^e} w_b^e M_b^e(\xi)}, \tag{2}$$

where N_a^e is the a-th rational T-spline blending function over the element e. This finalizes the process to construct the basis functions in parameter space. We would like to point out that if all the weights take the value one, then the rational T-spline blending functions become polynomials due to the partition of unity property, that is,

$$\sum_{b=1}^{n^e} M_b^e(\xi) = 1. \tag{3}$$

Finally, to perform computations on non-trivial geometries, we need to map the parent element to physical space using a geometrical mapping $x^e : \square \mapsto \Omega^e$. The basic information needed to produce this mapping is a set of geometry control points $\{Q_B\}_{B=1}^n$ and the set of weights we just introduced, namely, $\{w_B\}_{B=1}^n$. A given geometry control point Q_A will have local indices associated, such that $Q_A = Q_a^e$, where $A = \mathrm{IEN}(a, e)$. The same argument applies to weights. Thus, the geometrical mapping local to element e can be defined as

$$x^e(\xi) = \sum_{a=1}^{n^e} Q_a^e N_a^e(\xi) \quad \xi \in \square. \tag{4}$$

Using the T-spline geometrical mapping (4), we can obtain the basis functions in physical space. In particular, the restriction of N_A to the element e is just the push forward of N_a^e, where $A = \mathrm{IEN}(a, e)$. Equation (4) will also be used to map the elemental T-mesh to the Bézier physical mesh. Figure 3(b) shows a representative example of a Bézier mesh. In summary, as shown by Equations (1)–(4), the use of Bézier extraction allows to use T-splines in a finite element code modifying just the shape function subroutine.

DOI: 10.1002/nme

2.4. Continuity of T-spline blending functions

A T-spline blending function is at least $\mathcal{C}^{p-1}$ across an element boundary shared by two regular elements if the knot intervals associated to the edges in the other topological direction are not zero. The number of continuous derivatives would be decreased by m if we introduce m zero-length knot intervals. Now, we proceed to describe the continuity in the two-ring neighborhood of an extraordinary node. A T-spline blending function is $\mathcal{C}^2$-continuous across an element boundary shared by a regular and an irregular element. The continuity is reduced to $\mathcal{C}^0$ across an element boundary shared by two irregular elements in the one-ring of an extraordinary node. The continuity across element boundaries in the two-ring of an extraordinary node not described heretofore is $\mathcal{C}^1$.

2.5. Greville points

In our FSI algorithm presented in the following section, we will use a collocation approach that takes inspiration from the emerging field of isogeometric collocation [52, 56]. Isogeometric collocation is often carried out using the so-called Greville points. Greville points are easily defined for B-Splines and NURBS [91], but their extension to unstructured T-splines is not trivial. Here, we utilize the same generalization of classical Greville points that was used in [45]. We associate a Greville point to each vertex of the T-mesh. Greville points associated to extraordinary nodes are located right on top of extraordinary nodes. The location of Greville points associated to vertices which are not extraordinary is defined by the knot intervals assigned to the edges that touch that vertex (see the details in [45]). We can collect all Greville points defined this way in the set $\{\widehat{\tau}_i\}_{i=1}^n$. For each of these Greville points, we need to find the element in which it falls and its corresponding coordinates in the parent element, namely, ξ_i^e. The Greville points in physical space are computed as $\tilde{\tau}_i = x^e\left(\xi_i^e\right)$. The set of all those Greville points in physical space is denoted by $\widetilde{\mathcal{M}}^h = \{\widetilde{\tau}_i\}_{i=1}^n$. The Greville points for the T-mesh shown in Figure 1(a) are plotted in Figure 3(b) using small green squares.

2.6. Analysis-suitable T-splines

Without any restriction on the T-mesh topology, T-splines blending functions do not inherit all the important mathematical properties of NURBS. In particular, some problems can appear with linear independence and partition of unity. For T-meshes of rectangular topology, this issue was solved in [44] by defining a subset of T-splines called ASTS where all the mathematical properties of NURBS are guaranteed including linear independence and partition of unity. When the T-mesh is unstructured, assuring the linear independence and partition of unity of blending functions is more difficult. In what follows, we summarize the current understanding concerning analysis suitability of T-splines with and without extraordinary nodes:

1. It was proven in [44] that a T-mesh with T-junctions and no extraordinary nodes gives rise to ASTS if no horizontal and vertical T-junction extensions intersect with each other.
2. It was proposed in [45] that a T-mesh with extraordinary nodes and no T-junctions gives rise to ASTS if no extraordinary node lies within the three-ring neighborhood of another extraordinary node.
3. It was proposed in [45] that a T-mesh with T-junctions and extraordinary nodes gives rise to ASTS if the conditions (1) and (2) hold true and no one-bay face extension spans a face in the three-ring neighborhood of an extraordinary node.

Note that ASTS are defined using simple topological rules. Although it is possible to construct linear independent T-spline blending functions violating the aforementioned conditions, in what follows, we restrict ourselves to the set of ASTS.

3. IMMERSED FLUID-STRUCTURE INTERACTION METHOD

In this section, we describe the basics of our formulation at the continuous and discrete levels, and we focus on the main differences between working with T-splines and NURBS for this immersed

DOI: 10.1002/nme

FSI methodology. The detailed derivations of the formulation at the continuous and discrete levels can be found in our earlier paper [8].

3.1. Fluid–solid system

In what follows, $\Omega \subset \mathbb{R}^d$ is a time-independent open set which encloses our fluid–solid system. $\Omega_t^f \subset \Omega$ denotes the open subset of Ω occupied by the fluid at time t, which we call the fluid domain. The solid domain at time t is $\Omega_t^f \subset \Omega$. Note that, although we have assumed Ω to be fixed in time, the fluid and solid domains do depend on time, as indicated by the subscript. The solid and fluid domains define a partition of Ω such that $\overline{\Omega} = \overline{\Omega_t^f \cup \Omega_t^s}$. The fluid and solid domains are not allowed to overlap ($\Omega_t^f \cap \Omega_t^s = \emptyset$, where $\emptyset$ is the empty set), but they meet at the solid–fluid interface that we call Γ_t^I ($\overline{\Omega_t^f} \cap \overline{\Omega_t^s} = \Gamma_t^I$). The boundaries of Ω, Ω_t^f, and Ω_t^s are denoted by Γ, Γ_t^f, and Γ_t^s, respectively, and their unit outward normals by n, n^f, and n^s, where the subscript t is omitted for notational simplicity.

3.2. Kinematics

A Lagrangian description will be used for the solid domain, so let us define a reference configuration Ω_0^s for our solid body. We consider the mapping $\varphi : \Omega_0^s \times (0, T) \mapsto \mathbb{R}^d$, where $(0, T)$ is the time interval of interest. We assume that φ is sufficiently smooth, orientation preserving, and invertible [92]. Points $\vec{X}$ in Ω_0^s are called material points or particles, while points in $\mathbb{R}^d$ are denoted x and are called spatial points. To define the mapping φ, we make use of the displacement field $\overline{u}^s : \Omega_0^s \times (0, T) \mapsto \mathbb{R}^d$. In particular,

$$\varphi(X, t) = X + \overline{u}^s(X, t). \tag{5}$$

We define the velocity $\overline{v}^s : \Omega_0^s \times (0, T) \mapsto \mathbb{R}^d$, and the acceleration $\overline{a}^s : \Omega_0^s \times (0, T) \mapsto \mathbb{R}^d$ of a material point as

$$\overline{v}^s(X, t) = \frac{\partial \overline{u}^s(X, t)}{\partial t}, \tag{6}$$

$$\overline{a}^s(X, t) = \frac{\partial \overline{v}^s(X, t)}{\partial t} = \frac{\partial^2 \overline{u}^s(X, t)}{\partial t^2}. \tag{7}$$

Associated to the function $\overline{v}^s$, we define $v^s : \Omega_t^s \times (0, T) \mapsto \mathbb{R}^d$ as

$$\overline{v}^s(X, t) = v^s(\varphi(X, t), t) \quad \text{for all} \quad X \in \Omega_0^s,\ t \in [0, T]. \tag{8}$$

Note that although $\overline{v}^s$ and v^s represent the same physical quantity, namely, the solid velocity, they are different functions, and thus, we use different notation for them. In what follows, we will call $\overline{v}^s$ Lagrangian or material velocity and v^s Eulerian or spatial velocity. In general, we will use a bar for Lagrangian functions whenever there is possibility of confusion. Applying the chain rule to (8), we obtain

$$\overline{a}^s(X, t) = \frac{\partial \overline{v}^s}{\partial t}(X, t) = \frac{\partial v^s}{\partial t}(x, t) + v^s(x, t) \cdot \nabla_x v^s(x, t) \quad \text{if} \quad x = \varphi(X, t). \tag{9}$$

In the following, we will also use the notation

$$\dot{v^s} = \frac{\partial v^s}{\partial t} + v^s \cdot \nabla_x v^s, \tag{10}$$

which is standard in computational mechanics. From the mapping φ, we may also define the deformation gradient $F : \Omega_0^s \times (0, T) \mapsto \mathbb{R}^{d \times d}$ as

$$F = \nabla_X \varphi = I + \nabla_X \overline{u}^s. \tag{11}$$

DOI: 10.1002/nme

In (11), I denotes the identity tensor in $\mathbb{R}^{d\times d}$. We will also make use of the Green–Lagrange strain tensor $E : \Omega_0^s \times (0,T) \mapsto \mathbb{R}^{d\times d}_{\text{sym}}$ defined by

$$E = \frac{1}{2}\left(C - I\right), \tag{12}$$

where $C : \Omega_0^s \times (0,T) \mapsto \mathbb{R}^{d\times d}_{\text{sym}}$ is called Cauchy–Green deformation tensor and takes on the form $C = FF^T$.

3.3. Governing equations in strong form

In this paper, we assume that both the fluid and the solid are incompressible.[§] Therefore, the main unknowns of the problem will be a pressure field and the velocity. The main idea of the method is to define global functions $v : \Omega \times (0,T) \mapsto \mathbb{R}^d$, $p : \Omega \times (0,T) \mapsto \mathbb{R}$, such that

$$v = \begin{cases} v^f & \text{on} \quad \Omega_t^f \times (0,T) \\ v^s & \text{on} \quad \Omega_t^s \times (0,T) \end{cases}; \qquad p = \begin{cases} p^f & \text{on} \quad \Omega_t^f \times (0,T) \\ p^s & \text{on} \quad \Omega_t^s \times (0,T) \end{cases}; \tag{13}$$

where v^f and p^f are the fluid velocity and pressure in Eulerian coordinates, while v^s and p^s are analogous quantities for the solid. The pressure field will be used as a Lagrange multiplier to impose the incompressibility constraint which holds in both the solid and the fluid. The function v is globally continuous due to the no-slip condition at the fluid–solid interface

$$v^f = v^s \quad \text{on} \quad \Gamma_t^I \times (0,T), \tag{14}$$

but the global pressure could, in principle, be discontinuous. Using the functions v and p, taking into account that both fluid and solid are considered to be incompressible, and assuming that gravity is the only external force acting on the system, the linear momentum balance and mass conservation equations of our FSI problem can be written as follows

$$\rho^f \dot{v} = \nabla_x \cdot \sigma^f + \rho^f g + \boldsymbol{\mathcal{F}} \qquad \text{in} \quad \Omega \times (0,T), \tag{15}$$

$$\nabla_x \cdot v = 0 \qquad \text{in} \quad \Omega \times (0,T), \tag{16}$$

where $\mathcal{F}$ is defined as

$$\boldsymbol{\mathcal{F}} = \begin{cases} 0, & x \in \Omega_t^f \\ \left(\rho^f - \rho^s\right)\left(\dot{v} - \boldsymbol{g}\right) + \nabla_x \cdot \left(\boldsymbol{\sigma}^s - \boldsymbol{\sigma}^f\right), & x \in \Omega_t^s \end{cases}. \tag{17}$$

In Equations (15)–(17), ρ^f and σ^f denote the fluid density and Cauchy stress tensor, respectively, while ρ^s and σ^s denote the same quantities for the solid. The vector g denotes the acceleration of gravity. To guarantee correct transmission of forces at the fluid–solid interface, we need to impose the constraint

$$\sigma^f n^f = -\sigma^s n^s \quad \text{on} \quad \Gamma_t^I \times (0,T). \tag{18}$$

At this point, we need to introduce a constitutive theory to define the mechanical response of the fluid and the solid. We will assume the fluid to be Newtonian. Therefore,

$$\sigma^f = -pI + 2\mu \nabla_x^{\text{sym}} v, \tag{19}$$

where $\mu > 0$ is the dynamic viscosity and $\nabla_x^{\text{sym}} v = (\nabla_x v + \nabla_x v^T)/2$. Note that we have extended the definition of σ^f to the entire fluid-solid system, even if it does not have a clear physical meaning

[§]The reader interested in the necessary modifications to account for a compressible solid embedded in an incompressible fluid is remitted to [93].

DOI: 10.1002/nme

on Ω_t^s. We consider deformable solids in this paper.[¶] The Cauchy stress tensor of the solid is given by the expression

$$\sigma = -pI + FSF^T/\boldsymbol{J}, \tag{20}$$

where $J = \det(F)$ and S is the second Piola–Kirchhoff stress tensor. We will consider a hyperelastic material, in particular, the Neo-Hookean model with dilatation penalty proposed in [94]. For this particular material type, the second Piola–Kirchhoff stress tensor takes on the form

$$\boldsymbol{S} = \mu^s J^{-2/d}\left(\boldsymbol{I} - \frac{1}{d}\mathrm{tr}(\boldsymbol{C})\boldsymbol{C}^{-1}\right) + \frac{1}{2}\kappa^s\left(J^2 - 1\right)\boldsymbol{C}^{-1}, \tag{21}$$

where μ^s is the shear modulus and κ^s is the bulk modulus. Note that the last term of the preceding equation (called dilatational term) is zero if the solid is incompressible, but we maintain it in the formulation for reasons that will be made clear in Section 4.1. To compute the stress tensor S, we need the Lagrangian displacement of the solid $\overline{u}^s(X,t)$, which in our immersed method will be obtained using the equation

$$\frac{\partial \overline{u}^s(X,t)}{\partial t} = v(x,t) \quad \text{if} \quad x = \varphi(X,t) \quad \text{for all} \quad X \in \Omega_0^s, \tag{22}$$

which follows from the definition of the global velocity v.

To complete the definition of a well-posed initial/boundary-value problem, we need to impose suitable initial and boundary conditions. To define adequate boundary conditions, we split the boundary Γ into non-overlapping sets associated to the fluid boundary and the solid boundary. Thus, we define $\mathcal{B}^s = \Gamma_t^s \cap$ and $\mathcal{B}^f = \Gamma_t^s \cap$ such that $\Gamma = \overline{\mathcal{B}^s \cup \mathcal{B}^f}$ and $\emptyset = \mathcal{B}^s \cap \mathcal{B}^f$. Note that in general, $\mathcal{B}^s \neq$, because part of may be immersed into Ω due to the nature of the method. Now, we specify standard boundary conditions for a fluid-mechanics and solid-mechanics problem on $\mathcal{B}^s$ and $\mathcal{B}^f$, respectively. For consistency with our hypothesis about the time independence of Ω, we only consider homogeneous displacement boundary conditions on $\mathcal{B}^s$. On the fluid boundary $\mathcal{B}^f$, we assume the typical split between Dirichlet and Neumann boundary conditions, which can be expressed as $\mathcal{B}^f = \overline{\mathcal{B}_D^f \cup \mathcal{B}_N^f}$, with $\mathcal{B}_D^f \cap \mathcal{B}_N^f = \emptyset$. Mathematically, our boundary conditions may be expressed as

$$v = v_B \qquad \text{on} \quad \mathcal{B}_D^f, \tag{23}$$

$$\boldsymbol{\sigma}^f n^f = h \qquad \text{on} \quad \mathcal{B}_N^f, \tag{24}$$

$$\overline{u}^s = 0 \qquad \text{on} \quad \mathcal{B}^s, \tag{25}$$

where v_B is a given velocity and h is a prescribed traction vector. As initial conditions, we set an initial velocity for the entire fluid–solid system. In the solid domain, we also need to solve for displacements, so we have to set initial displacements on . Therefore, our initial conditions are given by

$$v(x,0) = v_I(x), \quad x \in \Omega, \tag{26}$$

$$\overline{u}^s(X,0) = 0, \qquad X \in \Omega_0^s. \tag{27}$$

In summary, the main three equations to be solved in our immersed strategy are the linear momentum balance Equation 15, the mass conservation Equation 16 and the kinematic relation between the Lagrangian displacement and the Eulerian velocity given by Equation (22).

[¶]For the inclusion of rigid bodies in a immersed technique such as the one presented in this manuscript, various changes need to be performed. The interested reader is remitted to [95].

DOI: 10.1002/nme

3.4. Linear momentum balance and mass conservation equations in weak form

As mentioned before, Equation (22) will be directly discretized in strong form, but the linear momentum balance Equation 15, and the mass conservation Equation 16 will be solved using a weighted residual formulation. Thus, before discretization, we need to derive a weak form of Equations (15) and (16). To do so, we follow a variational multiscale approach proposed in [8]. We will omit the details here, but the main idea is to split the velocity and pressure fields into coarse-scale and fine-scale components as $v = \widetilde{v} + v'$ and $p = \tilde{p} + p'$. Then, we multiply Equations (15) and (16) with weight functions $w = \widetilde{w} + w'$ and $q = \tilde{q} + q'$, integrate over the relevant domain, and perform basic manipulations (see [8] for a detailed derivation). Proceeding as indicated, we obtain the following variational problem: find v and p such that for all $\widetilde{w}$ and $\tilde{p}$

$$B^{ED}\left(\{\widetilde{w},\tilde{q}\},\{\widetilde{v}+v',\tilde{p}+p'\}\right) - L^{ED}\left(\widetilde{w}\right) + B^{LD}\left(\widetilde{w},\widetilde{v}\right) - L^{LD}\left(\widetilde{w}\right) = 0, \tag{28}$$

with

$$\begin{aligned} B^{ED}\left(\{\widetilde{w},\tilde{q}\},\{\widetilde{v}+v',\tilde{p}+p'\}\right) = &\left(\widetilde{w},\frac{\partial\widetilde{v}}{\partial t}+\widetilde{v}\cdot\nabla_x\widetilde{v}\right)_\Omega \\ &-(\nabla_x\cdot\widetilde{w},\tilde{p})_\Omega + \left(\nabla_x^{\text{sym}}\widetilde{w},2\nu\nabla_x^{\text{sym}}\widetilde{v}\right)_\Omega \\ &+(\tilde{q},\nabla\cdot\widetilde{v})_\Omega + \left(\widetilde{w},v'\cdot\nabla\widetilde{v}\right)_\Omega \\ &-\left(\nabla_x\widetilde{w},(\widetilde{v}+v')\otimes v'\right)_\Omega - \left(\nabla_x\cdot\widetilde{w},p'\right)_\Omega \\ &-\left(\nabla\tilde{q},v'\right)_\Omega, \end{aligned} \tag{29}$$

$$L^{ED}\left(\widetilde{w}\right) = \left(\widetilde{w}(x),\boldsymbol{g}\right)_\Omega + \left(\widetilde{w}(x),h(x)/\rho^f\right)_{\mathcal{B}_N^f}, \tag{30}$$

$$\begin{aligned} B^{LD}\left(\widetilde{w},\widetilde{v}\right) = &-\left(\widetilde{w}(\varphi(X,t)),\left(1-\frac{\rho^s}{\rho^f}\right)\frac{\partial\widetilde{v}(\varphi(X,t),t)}{\partial t}J(X,t)\right)_{\Omega_0^s} \\ &-\left(\widetilde{w}(\varphi(X,t)),\left(1-\frac{\rho^s}{\rho^f}\right)\widetilde{v}(\varphi(X,t),t)\cdot\nabla_x\widetilde{v}(\varphi(X,t),t)J(X,t)\right)_{\Omega_0^s} \\ &-\left(\nabla_x^{\text{sym}}\widetilde{w}(\varphi(X,t)),2\nu\nabla_x^{\text{sym}}\widetilde{v}(\varphi(X,t),t)J(X,t)\right)_{\Omega_0^s} \\ &+\left(\nabla_x^{\text{sym}}\widetilde{w}(\varphi(X,t)),\frac{1}{\rho^f}F(X,t)S(X,t)F^T(X,t)\right)_{\Omega_0^s}, \end{aligned} \tag{31}$$

$$L^{LD}\left(\widetilde{w}\right) = -\left(\widetilde{w}(\varphi(X,t)),\left(1-\frac{\rho^s}{\rho^f}\right)\boldsymbol{g}\,J(X,t)\right)_{\Omega_0^s}. \tag{32}$$

Here, $(\cdot,\cdot)$ denotes the $\mathcal{L}^2$ inner product over the domain indicated with the corresponding subscript, $\otimes$ denotes the outer product, and $\nu = \mu^f/\rho^f$ is the kinematic viscosity. The superscript ED emphasizes that the equation is defined in the Eulerian description, whereas the superscript LD indicates the use of the Lagrangian description.

3.5. Space discretization

We apply a semi-discrete formulation to our FSI problem. Therefore, we first discretize in space, reducing Equation (28) to a system of ordinary differential equations that will be eventually integrated in time. To perform space discretization, we need to introduce two discrete spaces associated to the Eulerian and Lagrangian unknowns. In both cases, our discrete spaces will be spanned by

DOI: 10.1002/nme

linearly independent T-spline basis functions constructed as indicated in Section 2. Let us introduce the set of T-spline basis functions $\{N_A(x)\}_{A=1}^{n_{ED}}$, which is defined over the physical space Ω and is employed to discretize Eulerian unknowns. Similarly, we define the set of T-spline basis functions $\{\overline{N}_B(X)\}_{B=1}^{n_{LD}}$ over the reference configuration of the solid Ω_0^s. Assuming that Dirichlet boundary conditions will be strongly enforced on the discrete space at a later stage, we can write

$$v^h(x,t) = \sum_{A=1}^{n_{ED}} v_A(t) N_A(x); \quad p^h(x,t) = \sum_{A=1}^{n_{ED}} p_A(t) N_A(x); \tag{33}$$

$$w^h(x) = \sum_{A=1}^{n_{ED}} w_A N_A(x); \quad q^h(x) = \sum_{A=1}^{n_{ED}} q_A N_A(x); \tag{34}$$

$$\overline{u}^h(X,t) = \sum_{B=1}^{n_{LD}} \overline{u}_B(t) \overline{N}_B(X); \tag{35}$$

where v^h, p^h, w^h, q^h, and $\overline{u}^h$ are the discrete counterparts of $\widetilde{v}$, $\tilde{p}$, $\widetilde{w}$, $\tilde{q}$, and $\overline{u}^s$, respectively. A discrete mapping is also defined $\varphi^h(X,t)$, which will be used to push forward the Lagrangian mesh to physical space. We will now use the isoparametric concept, that is, we will use the same shape functions to represent the geometry. Thus, the basis functions $\{N_A(x)\}_{A=1}^{n_{ED}}$ are used to define a computational mesh on the whole domain Ω. We will refer to this mesh as the Eulerian mesh, and it will be utilized to compute the integrals of the weak form (28) on Ω and $\mathcal{B}_N^f$. Analogously, the integrals of the weak form (28) on will be computed on the mesh defined by the basis functions $\{\overline{N}_B(X)\}_{B=1}^{n_{LD}}$. This second mesh, because it is defined in the material description, will be referred to as the Lagrangian mesh. Note that the Lagrangian mesh is arbitrarily located on top of the Eulerian mesh.

Finally, to compute Equation (28), we need a model for the fine-scale velocity v' and fine-scale pressure p'. Here, we use the standard approach

$$v' = -\tau_M r_M \left(v^h, p^h\right), \tag{36}$$

$$p' = -\tau_C r_C \left(v^h\right), \tag{37}$$

where τ_M and τ_C are the stabilization parameters defined in [10], and

$$r_M \left(v^h, p^h\right) = \frac{\partial v^h}{\partial t} + v^h \cdot \Delta_x v^h + \Delta_x p^h - \nu \Delta_x v^h - g, \tag{38}$$

$$r_C \left(v^h\right) = \Delta_x . v^h. \tag{39}$$

Replacing continuous with discrete variables in Equation (28) and using (36)–(37), we obtain the final form of our semi-discrete weak formulation, which may be expressed as follows: find v^h and p^h such that, for all w^h and q^h defined as in Equation (34),

$$B_{MS}^{ED}\left(\left\{w^h, q^h\right\}, \left\{v^h, p^h\right\}\right) - L^{ED}\left(w^h\right) + B^{LD}\left(w^h, v^h\right) - L^{LD}\left(w^h\right) = 0, \tag{40}$$

with

DOI: 10.1002/nme

$$B_{MS}^{ED}\left(\left\{w^h,q^h\right\},\left\{v^h,p^h\right\}\right)=\left(w^h(x),\frac{\partial v^h(x,t)}{\partial t}+v^h(x,t)\Delta_x v^h(x,t)\right)_\Omega$$
$$-\left(\Delta_x.\,w^h(x),p^h(x)\right)_\Omega+\left(\nabla_x^{\mathrm{sym}}w^h(x),2\nu\nabla_x^{\mathrm{sym}}v^h(x,t)\right)_\Omega$$
$$+\left(q^h(x),\Delta_x.\,v^h(x,t)\right)_\Omega-\left(w^h(x),\tau_M r_M\left(u^h,p^h\right)\cdot\nabla v^h\right)_\Omega$$
$$+\left(\Delta_x w^h(x),\left(v^h-\tau_M r_M\left(v^h,p^h\right)\right)\otimes\tau_M r_M\left(v^h,p^h\right)\right)_\Omega$$
$$+\left(\Delta_x.\,w^h(x),\tau_C r_C\left(u^h\right)\right)_\Omega+\left(\Delta_x\,q^h,\tau_M r_M\left(u^h,p^h\right)\right)_\Omega. \tag{41}$$

This completes the space discretization of the linear momentum balance and the mass conservation equations. To finalize the space discretization, we also need to discretize Equation (22). As mentioned before, we will discretize Equation (22) directly in strong form using a collocation approach. Thus, we replace Equation (22) with a set of discrete equations given by

$$\frac{\partial\overline{u}^h}{\partial t}(\tilde{\tau}_i,t)=v^h(\varphi^h(\tilde{\tau}_i,t),t)\quad\text{for all}\quad\tilde{\tau}_i\in\widetilde{\mathcal{M}}^h, \tag{42}$$

where $\widetilde{\mathcal{M}}^h$ is the set of Greville points in physical space as defined in Section 2. Note that Equations (40) and (42) are fully coupled and will be solved simultaneously using a monolithic approach. Associated to Equations (40) and (42), we define the following residual vectors

$$R^M=\left\{R^M_{A,i}\right\};\quad R^C=\left\{R^C_A\right\};\quad R^P=\left\{R^P_{B,i}\right\}; \tag{43}$$

where $A\in\{1,\ldots,n_{ED}\}$ is a control-variable index of the Eulerian mesh, $B\in\{1,\ldots,n_{LD}\}$ is a control-variable index of the Lagrangian mesh and i is a dimension index which goes from 1 to d. The components of the residual vectors have the following expressions

$$R^M_{A,i}=B_{MS}^{ED}\left(\{N_Ae_i,0\},\{v^h,p^h\}\right)-L^{ED}\left(N_Ae_i\right)+B^{LD}\left(N_Ae_i,v^h\right)-L^{LD}\left(N_Ae_i\right), \tag{44}$$

$$R^C_A=B_{MS}^{ED}\left(\{0,N_A\},\{v^h,p^h\}\right), \tag{45}$$

$$R^P_{B,i}=e_i\cdot\left[\frac{\partial\overline{u}^h}{\partial t}(\tilde{\tau}_B,t)-v^h\left(\varphi^h(\tilde{\tau}_B,t),t\right)\right], \tag{46}$$

where e_i is the ith vector of the Cartesian basis.

3.6. Information transfer between the Eulerian and the Lagrangian meshes

Equation (46) connects the Lagrangian displacement and the Eulerian velocity at locations given by the Greville points $\{\tilde{\tau}_B\}_{B=1}^{n_{LD}}$, which live in the solid's reference configuration. These points have been computed from their parametric counterparts $\{\widehat{\tau}_B\}_{B=1}^{n_{LD}}$, which are known and given by the parametric configuration of the Lagrangian mesh. Thus, the term involving $\overline{u}^h$ in Equation (46) can be readily evaluated using typical procedures of the finite element method. The same argument does not apply to the term involving v^h because it is defined in the Eulerian mesh. According to Equation (46), v^h has to be evaluated in the physical point $x_B=\varphi^h(\tilde{\tau}_B,t)$. We know the parametric origin of x_B in the Lagrangian mesh, that is $\widehat{\tau}_B$, but what we need to compute $v^h(\varphi^h(\tilde{\tau}_B,t),t)$ in a standard finite element fashion is its parametric origin in the Eulerian mesh, which is not known *a priori*. Thus, to evaluate $v^h(\varphi^h(\tilde{\tau}_B,t),t)$ in Equation (46), we need to invert the push forward

DOI: 10.1002/nme

to the physical space of the Eulerian mesh. Note that a similar issue arises when computing the operators B^{LD} and L^{LD} in Equation (44). These operators involve integrals over the Lagrangian domain that require evaluations of Eulerian functions on quadrature points of the Lagrangian mesh. The parametric origin of these points in the Eulerian mesh is not known either.

Our computations are based on Bézier extraction. Therefore, the mapping from parametric to physical space is local to elements as shown in Equation (4). As a consequence, if we want to compute the parametric point associated to x_B in the Eulerian mesh, we first need to find the Bézier element in which x_B is located. Bézier elements, in general, will have curvilinear shape, which makes it difficult to locate the element without inverting the mapping. Thus, the most general way to state the problem mathematically is find $e \in \{1, \ldots, N_{el}\}$ and $\xi_B \in \square$ such that

$$x_B = \sum_{a=1}^{n^e} Q_a^e N_a^e(\xi_B), \tag{47}$$

where N_{el} is the global number of Bézier elements. Solving Equation (47) for all Bézier elements would be computationally very intensive. Therefore, in practical dynamic computations, we proceed as follows: For a given point $\tilde{\tau}_B$, we store the Bézier element in which its physical counterpart fell at the previous time step. When we have to find the Bézier element in which it falls in the current time step, we only solve Equation (47) for a very small set of elements. That reduced set of elements is composed by that which hosted the point in the previous time step and their neighbors.

Remark:

1. The local geometrical mapping is inverted solving a $d \times d$ nonlinear system using the Newton–Raphson algorithm.
2. When inverting the local geometrical mapping, it is important to check that the solution ξ_B actually satisfies $\xi_B \in \square$. As a matter of fact, Equation (47) usually has roots outside the parent element and they need to be disregarded.
3. The strategy to transfer information between the Eulerian and the Lagrangian meshes is simpler when using NURBS rather than T-splines. In NURBS-based analysis, the geometrical mapping is local to patches rather than elements. Therefore, the first step in the process is to find the patch that hosts the physical point. Because the number of patches is typically much lower than the number of elements, the process is simplified. In addition, in many cases of practical interest, NURBS geometrical mappings can be inverted analytically. Within this latter category, we can include computations on boxes, cylinders or spheres, for example.

3.7. Time integration

Once space discretization has been sorted, we must now proceed with time discretization. A monolithic and fully implicit algorithm based on the generalized-α method was proposed by some of the authors in [8] and will be used here. The generalized-α algorithm dates back to 1993, when it was first proposed to solve the structural mechanics equations [85]. The method was later extended to first-order systems in the fluid dynamics domain [86]. As usual, we divide the time interval of interest $[0, T]$ into a sequence of subintervals (t_n, t_{n+1}) with fixed time-step size $\Delta t = t_{n+1} - t_n$. In the following, the time-discrete approximation at t_n of the global vectors of control variables of v^h, p^h, and $\frac{\partial v^h}{\partial t}$ will be referred to as V_n, P_n, and A_n, respectively. Analogously, the time-discrete approximation at t_n of the global vectors of control variables of $p^h v^h$ and $\frac{\partial v^h}{\partial t}$ will be $\overline{U}_n$ and $\overline{V}_n$. We will also introduce $\overline{U}_n^G$ as the global vector that collects the time-discrete approximation of the solid displacements at the Greville points, that is, $\overline{U}_n^G \approx \{\overline{u}_i^h(\tilde{\tau}_A, t_n)\}_{A=1}^{n_{LD}}$ for all $i = 1, \ldots, d$. In a similar manner, we define $\overline{V}_n^G \approx \{\frac{\partial \overline{u}_i^h}{\partial t}(\tilde{\tau}_A, t_n)\}_{A=1}^{n_{LD}}$ for all $i = 1, \ldots, d$. Following this notation, our time-integration algorithm may be defined as follows: Given V_n, A_n, $\overline{U}_n$, and $\overline{V}_n$, find V_{n+1}, A_{n+1}, $V_{n+\alpha_f}$, $A_{n+\alpha_m}$, P_{n+1}, $\overline{U}_{n+1}$, $\overline{V}_{n+1}$, $\overline{U}_{n+\alpha_f}$, and $\overline{V}_{n+\alpha_m}$ such that

$$R^M\left(\boldsymbol{V}_{n+\alpha_f}, \boldsymbol{A}_{n+\alpha_m}, \boldsymbol{P}_{n+1}, \overline{U}_{n+\alpha_f}\right) = 0, \tag{48}$$

DOI: 10.1002/nme

$$R^C(\boldsymbol{V}_{n+\alpha_f}, \boldsymbol{A}_{n+\alpha_m}, \boldsymbol{P}_{n+1}) = 0, \tag{49}$$

$$R^P\left(\overline{U}^G_{n+\alpha_f}, \overline{V}^G_{n+\alpha_m}, \boldsymbol{V}_{n+\alpha_f}\right) = 0, \tag{50}$$

$$\boldsymbol{V}_{n+\alpha_f} = \boldsymbol{V}_n + \alpha_f(\boldsymbol{V}_{n+1} - \boldsymbol{V}_n), \tag{51}$$

$$\boldsymbol{A}_{n+\alpha_m} = \boldsymbol{A}_n + \alpha_m(\boldsymbol{A}_{n+1} - \boldsymbol{A}_n), \tag{52}$$

$$\overline{U}^G_{n+\alpha_f} = \overline{U}^G_n + \alpha_f\left(\overline{U}^G_{n+1} - \overline{U}^G_n\right), \tag{53}$$

$$\overline{V}^G_{n+\alpha_m} = \overline{V}^G_n + \alpha_m\left(\overline{V}^G_{n+1} - \overline{V}^G_n\right), \tag{54}$$

$$\boldsymbol{V}_{n+1} = \boldsymbol{V}_n + \Delta t((1-\gamma)\boldsymbol{A}_n + \gamma\boldsymbol{A}_{n+1}), \tag{55}$$

$$\overline{\boldsymbol{U}}^G_{n+1} = \overline{\boldsymbol{U}}^G_n + \Delta t\left((1-\gamma)\overline{\boldsymbol{V}}^G_n + \gamma\overline{\boldsymbol{V}}^G_{n+1}\right), \tag{56}$$

where α_m, α_f, and γ are real-valued parameters that control the accuracy and stability of the algorithm. We will choose here the values proposed by Jansen *et al.* [86] for first-order ordinary differential equations. Note also that the vectors $\overline{U}_{n+\alpha_f}$ and $\overline{U}^G_{n+\alpha_f}$ are related through a linear equation. In fact, to obtain the vector of control variables from the values at Greville points, we just need to collocate Equation (35) at Greville points as

$$\overline{u}^h(\tilde{\tau}_i, t_{n+\alpha_f}) = \sum_{B=1}^{n_{LD}} \overline{u}_B(t_{n+\alpha_f})\overline{N}_B(\tilde{\tau}_i) \quad \text{for all} \quad \tilde{\tau}_i \in \widetilde{\mathcal{M}}^h. \tag{57}$$

Equation (57) represents a linear system that establishes a correspondence between $\overline{U}_{n+\alpha_f}$ and $\overline{U}^G_{n+\alpha_f}$. The matrix associated to (57) depends on the parametric configuration of the Lagrangian mesh only, making it time independent. Therefore, we just need to compute this matrix once and store it, then we will use it in each Newton–Rapshon of each time step.

Note that (50) is solved for each Greville point independently, that is, we do not need to assemble any matrix for the kinematic equation at each Newton–Raphson iteration of every time step. This is one of the consequences of using our hybrid variational-collocation method. It increases the global efficiency of the methodology by saving computational time especially for higher orders.

4. NUMERICAL EXAMPLES

In this section, we present two numerical examples that illustrate the accuracy, stability, and robustness of our algorithm. The examples also compare global refinement (NURBS) against local refinement (T-splines) showing the enhanced efficiency of ASTS. The code used to perform these simulations has been developed on top of the scientific library PETSc [96].

4.1. Free-falling cylinder in a channel

Let us consider a channel of infinite length, infinite height, and width $2L$, which contains a viscous fluid. We assume that there is a solid cylinder of infinite length and radius a immersed in the fluid. The solid is denser than the surrounding fluid, and it is initially at rest. Then, it is allowed to fall freely under the action of gravity. When the solid is released, it will accelerate until the effect of the drag force, which becomes more prominent with velocity, added to buoyancy counterbalance the downward pull of gravity. At that point, the net force acting on the solid will be zero, and the

DOI: 10.1002/nme

Table I. Values of the relevant physical parameters for the free-falling cylinder benchmark problem.

Physical parameter	Value and units
g	981 cm/s^2
ρ^f	1.0 g/cm^3
μ	5 dyn/cm^2
ρ^s	1.25 g/cm^3
μ^s	40000 g/(cm s^2)
L	2 cm
a	0.25 cm

object will move with constant velocity. This velocity is known as the terminal or settling velocity v_T. If we consider the cylinder diameter to be the characteristic length-scale of the problem and the aforementioned terminal speed to be the characteristic velocity, then the relevant Reynolds number is

$$R_e = \frac{2\rho^f v_T a}{\mu}. \tag{58}$$

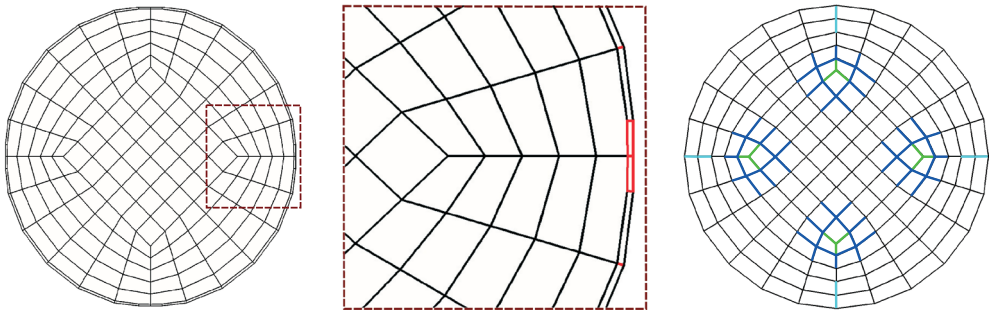
It may be shown that, under the assumption of creeping flow ($R_e << 1$), the terminal velocity may be approximated by the expression

$$v_T = \frac{\left(\rho^s - \rho^f\right) g a^2}{4\mu}\left[\ln\left(\frac{L}{a}\right) - 0.9157 + 1.7244\left(\frac{a}{L}\right)^2 - 1.7302\left(\frac{a}{L}\right)^4\right], \tag{59}$$

where g is the Euclidean norm of g. In this numerical example, we perform a full-scale FSI simulation to compute the solid motion and compare the numerical approximation to the terminal velocity with its theoretical value as given by Equation (59). The infinite length of the cylinder and the channel allows us to solve this problem in a two-dimensional domain of width $2L$. In the direction of gravity, we take a computational domain which is large enough to produce negligible errors due to its finite size. Our data indicate that the computational domain $\Omega = [0, 2L] \times [0, 4L]$ is sufficiently large to reproduce accurately the results of the free-falling cylinder. In the bottom and lateral boundaries of the box, we impose no-slip conditions, while on the top boundary we prescribe stress-free conditions. In our formulation, the cylinder is modeled as a deformable solid, but we use sufficiently large stiffness moduli in the constitutive theory (Equation (21)) so that strains are negligible, that is, the solid will just translate in vertical direction as a rigid body. Note that, even if we enforce incompressibility in the formulation, the dilatational term has also been included in (21). Consistent with previous findings by other authors [82], we have noticed that non-negligible errors in the incompressibility constraint arise if the dilatational term is dropped. We have followed [82] in order to fix this issue, that is, we have used relatively small values of κ^s for our computations (of the order of $\mu^s/10$) in order to maintain J close enough to one. The values of the physical parameters used in the computations are indicated in Table I. Introducing these parameters in Equation (59), we obtain $v_{\mathrm{T}} = 0.9122$ cm/s and the Reynolds number becomes $R_e = 0.09122$, which falls well within the creeping-flow regime.

The time step used in the simulations is $\Delta t = 10^{-3}$ s. We have discretized the problem in space using cubic T-spline shape functions. T-splines, unlike Lagrange piecewise polynomials, can represent the axial section of the cylinder exactly. Additionally, the unstructured nature of T-splines allows such an exact representation without the need to introduce singular points in the parameterization. We note that this is not possible using single-patch NURBS. We use the same Lagrangian mesh in all the simulations of this Section, it is plotted in Figure 4 along with its T-mesh. The shape functions of the Lagrangian mesh are at least $\mathcal{C}^2$-continuous everywhere except in some Bézier-element boundaries of reduced continuity, which are indicated in Figure 4 (see the caption of

DOI: 10.1002/nme



(a) Unstructured T-mesh (b) Detail of the T-mesh (c) Bézier mesh

Figure 4. Solid's T-mesh and Bézier mesh for the free-falling cylinder example. (a) T-mesh with four extraordinary nodes which allow us to avoid degenerated points in the parameterization. (b) Zoom-in of the T-mesh close to one of the extraordinary nodes. The color code in the edges of the T-mesh defines the knot interval configuration. All edges have a knot interval of 1 associated, except those colored in red which have associated a zero-length knot interval. (c) Bézier mesh in physical space. The basis functions are $\mathcal{C}^{\infty}$ everywhere, except at the boundaries of the Bézier elements, where the continuity is reduced. The continuity across black Bézier-element boundaries is at least $\mathcal{C}^2$. Dark blue lines represent element boundaries of $\mathcal{C}^1$ continuity due to the presence of extraordinary nodes. Green and light-blue lines represent element boundaries of $\mathcal{C}^0$ continuity due to the presence of extraordinary nodes and our choice to use exact geometry, respectively.

Table II. Different Eulerian meshes considered for the free falling cylinder benchmark problem.

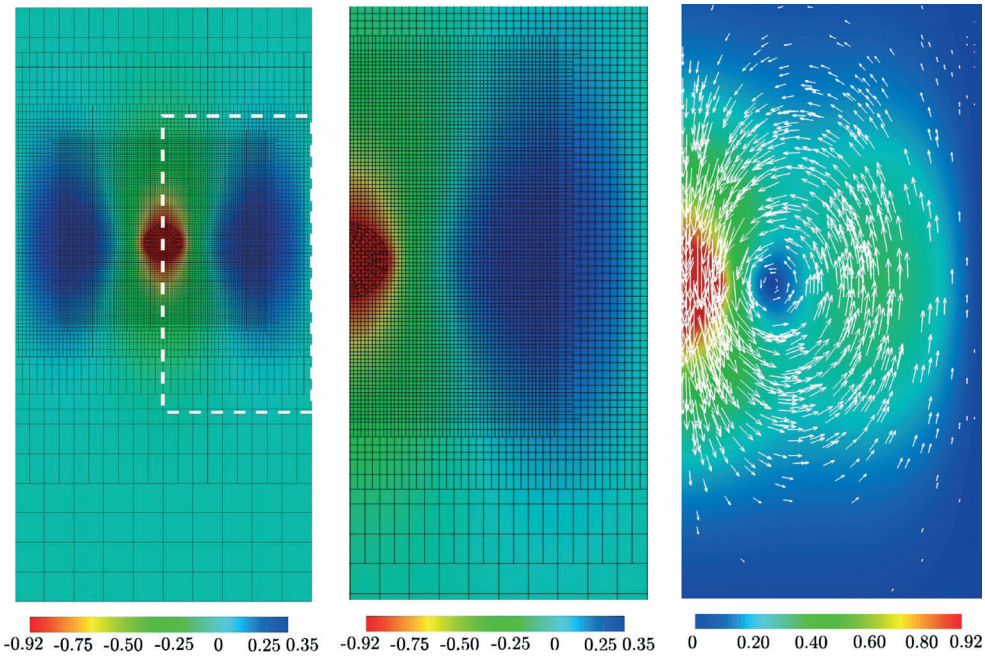
Eulerian mesh	Refinement	Smallest element size	Total number of Bézier elements
ASTS1	Three levels of refinement	0.1	1162
NURBS1	Uniform	0.1	3200
ASTS2	Four levels of refinement	0.05	3868
NURBS2	Uniform	0.05	12800
ASTS3	Five levels of refinement	0.025	14428
NURBS3	Uniform	0.025	51200

Figure 4 for details). We consider three ASTS Eulerian meshes and three NURBS Eulerian meshes. These meshes are depicted in Table II. Note that the size of the elements in each uniform NURBS mesh is always picked equal to the size of the smallest elements in its corresponding ASTS mesh. This is performed in order to make a fair comparison between local and global refinement. The shape functions of all Eulerian meshes are $\mathcal{C}^2$-continuous globally.

The entirety of the ASTS Eulerian mesh with five levels of refinement is shown in Figure 5 (left panel) superimposed with a color plot of the vertical velocity and the Lagrangian mesh. The area marked with a dashed white line (left panel) has been zoomed in for a clearer observation of the mesh and the solution (middle and right panels). The color code in Figure 5 indicates the vertical velocity (left and middle panels) and the velocity magnitude (right panel) at time $t = 0.75$ s. In the right panel, we have also plotted velocity vectors superimposed on the color plot. It is apparent that there is a large vortex close to the solid.

We have also computed the time evolution of the average vertical velocity of the cylinder for each Eulerian mesh. In Figure 6, it can be seen how the numerical terminal velocity gets closer to its theoretical value as we increase the spatial resolution for both ASTS and NURBS meshes. The globally and locally refined meshes with the same element size give essentially the same solution which shows the potential of ASTS in order to enhance the algorithm's efficiency. The relative error for the meshes ASTS3 and NURBS3 is below 0.2%, which seems to be similar to the error in the theoretical solution given the number of digits employed in the constants in Equation (59). Note that the mesh ASTS3 has 72% less Bézier elements than the mesh NURBS3 which exhibits the importance of local refinement. Additionally, we would like to point out that only 20 Bézier elements are behind the solid for the meshes ASTS1 and NURBS1, which is clearly not enough to give a good solution.

DOI: 10.1002/nme



(a) Vertical velocity (b) Vertical velocity (zoomed in) (c) Velocity magnitude (zoomed in)

Figure 5. (a) Eulerian and Lagrangian meshes along with a color plot of the vertical velocity. The Eulerian mesh is locally refined where the flow is most complex. Local refinement permitted us to reduce by 68% the number of Bézier elements with respect to a uniform mesh that achieves the same accuracy. (b) Zoom in of the inset area in the left panel. (c) Velocity magnitude in the inset area. The color plot is superimposed with velocity vectors, which show the formation of a large vortex close to the solid.

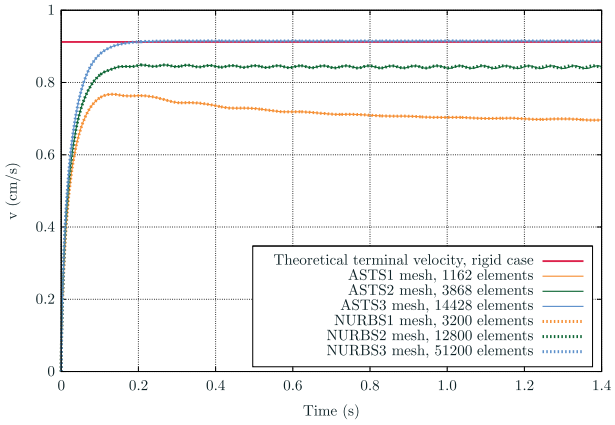


Figure 6. Time evolution of the average cylinder velocity for each Eulerian mesh compared against the theoretical terminal velocity (red line). The Lagrangian mesh is the same in all simulations and it is composed of 156 Bézier elements and 209 control points. The ASTS Eulerian meshes have been locally refined where the physics is more interesting without perceptible lost of accuracy from their globally refined NURBS meshes counterparts. Excellent agreement is obtained with respect to the theoretical terminal velocity when we use enough spatial resolution. Cubic basis functions have been used for all the meshes.

A peculiar feature of immersed FSI methods that use Lagrangian meshes arbitrarily embedded into a background Eulerian mesh is that the finite element spaces ensued from the meshes might have to comply with an inf-sup type condition to achieve stability [97, 98]. In the early stages of the IFEM, which is a particular case of our method for classical $\mathcal{C}^0$ finite elements, it was reported that the elements of the Lagrangian mesh had to be at least two times smaller than those of the Eulerian

DOI: 10.1002/nme

mesh. In our earlier work using globally $\mathcal{C}^1$-continuous basis functions, we showed that we could obtain stable and accurate results working with the same element size in both meshes [8]. Here, we show that the adoption of globally $\mathcal{C}^2$-continuous basis functions permits the use of solid elements two times larger than those of the background Eulerian mesh (see the meshes in the middle panel of Figure 5). These data suggest that increasing the global continuity may alleviate the inf-sup-type condition that the solid and fluid spaces have to satisfy. Although we acknowledge that we do not have a mathematical proof of this statement at this point, we do believe that this observation warrants further investigation.

Another important aspect is that the computation of the residual requires integrating on the Lagrangian mesh fields which live in the space defined by the Eulerian mesh. This leads to suboptimal quadrature because we are integrating on the elements of the Lagragian mesh, but some of the fields we integrate have lines of reduced continuity in those integration regions [99]. In classical finite elements, those lines have $\mathcal{C}^0$ continuity. However, the continuity of those lines is increased to $\mathcal{C}^2$ using cubic T-splines in the Eulerian mesh. It seems obvious that increasing the regularity of those lines of reduced continuity will lead to more accurate integration and better results.

In all, this example shows that, for this particular type of algorithms, it is important to be able to use local refinement *and* higher order global continuity simultaneously, which is very difficult to achieve without the use of T-spline-based isogeometric analysis.

4.2. Flow past a deformable solid with one end fixed

Let us consider the two-dimensional fluid–solid system depicted in Figure 7, where lengths are expressed in centimeters. The green rectangle in the plot represents a deformable solid of size 0.8×0.2 cm. The bottom of the solid is fixed to the channel wall. The system is driven by a uniform inflow velocity at the left boundary of the domain. The outlet on the right-hand side is subjected to a zero-traction boundary condition, and the top and bottom boundaries are assumed to satisfy no-slip boundary conditions. The values of the relevant physical parameters are indicated in Table III. We assume that the relevant Reynolds number for this problem is

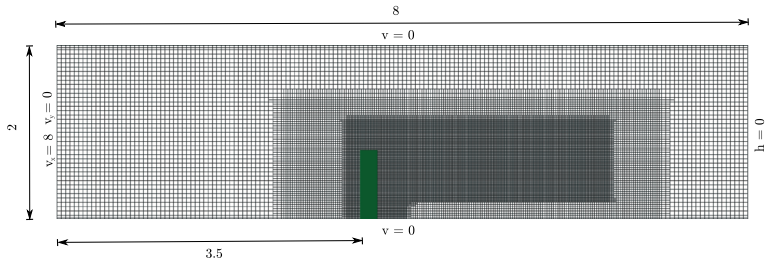


Figure 7. Flow past a solid. Geometry, boundary conditions, initial position of the solid, and Eulerian mesh. The introduction of T-junctions allows us to locally refine the Eulerian mesh at the boundary layers and downstream the solid. There are three levels of refinement. The solid is a rectangle of size 0.2×0.8 cm and its bottom end is fixed. The solid mesh is uniform. The lengths marked in the plot are expressed in centimeters and the velocities in centimeters per second.

Table III. Values of the relevant physical parameters for the flow past a solid example.

Physical parameter	Value and units
g	0 cm/s^2
ρ^f	1.0 g/cm^3
μ	0.02 dyn/cm^2
ρ^s	1.5 g/cm^3
μ^s	8000 g/(cm s^2)
v_{if}	8 cm/s
b	2 cm

DOI: 10.1002/nme

$$R_e = \frac{\rho^f \, b \, v_{if}}{\mu}, \tag{60}$$

where v_{if} is the inflow velocity and b is the length of the inflow boundary of the channel. As shown in Figure 7 and Table III, $b = 2$ cm. Using the remaining data in Table III, we find that $R_e = 800$.

As in the aforemetioned example, three cubic ASTS and NURBS Eulerian meshes are considered in order to do comparisons. The ASTS Eulerian meshes are locally refined where the flow is supposed to be more complex due to the presence of the solid. Three levels of refinement are introduced in all ASTS meshes with different spatial resolutions. The main features of the ASTS and NURBS Eulerian meshes are described in Table IV. The mesh ASTS3 is shown in Figure 7. The Lagrangian mesh used in our simulation is uniform, with 25×100 Bézier elements and 2884 control points.

Table IV. Different Eulerian meshes considered for the flow past a solid example.

Eulerian mesh	Refinement	Smallest element size	Total number of Bézier elements
ASTS1	Three levels of refinement	0.05	2154
NURBS1	Uniform	0.05	6400
ASTS2	Three levels of refinement	0.025	7675
NURBS2	Uniform	0.025	25600
ASTS3	Three levels of refinement	0.0125	29549
NURBS3	Uniform	0.0125	102400

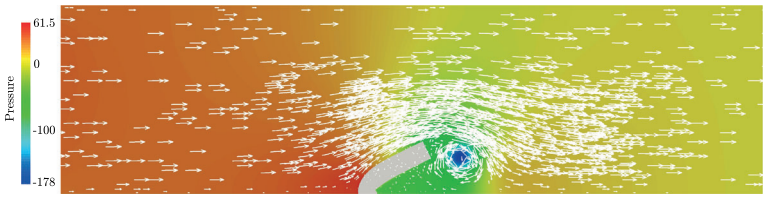


Figure 8. Flow past a solid. Pressure field color plot superimposed with arrows that represent the velocity at time $t = 0.193$ s. The plot clearly shows a vortex downstream the solid which coincides with the lowest-pressure area.

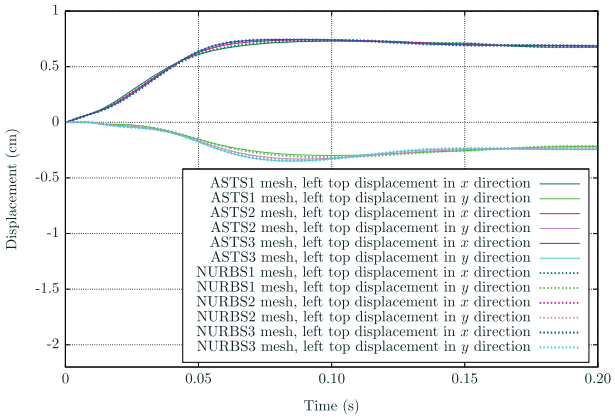


Figure 9. Flow past a solid. Time evolution of the displacement in the x and y directions of the left top corner of the solid for the different Eulerian meshes considered. The Lagrangian mesh is the same in all simulations and it is composed of 2500 Bézier elements and 2884 control points. The ASTS Eulerian meshes have been locally refined close and after the solid. The meshes NURBS2 and NURBS3 give very similar results despite of the much higher spatial resolution used in NURBS3, which indicates that enough spatial resolution has been used in the mesh NURBS3. The mesh ASTS3 provides basically the same displacement curves than NURBS3, however, it has 71% less Bézier elements thanks to the use of local refinement.

DOI: 10.1002/nme

The basis functions associated to both meshes are $\mathcal{C}^2$-continuous globally. The time step used in the simulation is $\Delta t = 2 \cdot 10^{-4}$ s.

Figure 8 shows the solution at time $t = 0.193$ s using the mesh ASTS3. The deformation of the solid is apparent. The color scale on the fluid represents the pressure and the superimposed arrows correspond to the velocity field. As expected, there is a large vortex downstream the solid, located in the low-pressure area. Figure 9 shows the time evolution of the horizontal and vertical displacements of the top left corner of the solid for different Eulerian meshes. The displacement curves barely vary from NURBS2 to NURBS3, which suggests that NURBS3 is a converged result. The mesh ASTS3 gives essentially the same solution than the mesh NURBS3, but it has 71% less Bézier elements which evidences the advantages of the local refinement capability of T-splines. Finally, the results of Figure 9 suggest that a steady configuration has been reached.

5. CONCLUSIONS

We proposed a hybrid variational-collocation immersed fluid-structure interaction method which uses T-spline-based isogeometric analysis for both the fluid and the solid. The information transfer between the Eulerian and the Lagrangian mesh is carried out using the T-splines shape functions as well. T-splines bring a number of possibilities to the fluid-structure interaction field thanks to their local refinement capability and their unstructured nature which allows to represent complex geometries. At the same time, T-splines maintain all the advantages that NURBS have against other types of shape functions such as high inter-element continuity. Higher-order global continuity was shown to increase the accuracy per degree of freedom in solid and fluid mechanics, and we showed that it is also beneficial for immersed FSI. Indeed, increasing the inter-element continuity of the discrete spaces alleviates previously reported constraints on the element size of the Lagrangian and Eulerian meshes. We have validated our computational framework solving a common FSI benchmark problem with theoretical solution. We took advantage of T-junctions to perform local refinement and extraordinary nodes to avoid singular points. Our results were in very good agreement with theoretical solutions, which confirms the feasibility of our methodology.

ACKNOWLEDGEMENTS

This work was accomplished during a 4-month visit of H. C. to Carnegie Mellon University (CMU) in 2014, for which financial support from the PhD student grant UDC-Inditex is gratefully acknowledged. H. G., H. C. and C. B-C. were partially supported by the European Research Council through the FP7 Ideas Starting Grant (project no. 307201). H. G. was also partially supported by Ministerio de Economía y Competitividad (contract no. DPI2013-44406-R) cofinanced with FEDER funds. L. L. and Y. Z. were partially supported by the PECASE Award N00014-14-10234. We acknowledge the open source scientific library PETSc and their developers.

REFERENCES

1. Hughes TJR, Cottrell JA, Bazilevs Y. Isogeometric analysis CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computacional Methods in Applied Mechanics and Engineering* 2005; **194**:4135–4195.
2. Cottrell JA, Hughes TJR, Bazilevs Y. *Isogeometric Analysis Toward Integration of CAD and FEA* Wiley (ed.). Wiley, 2009.
3. Bazilevs Y, Calo VM, Hughes TJR, Zhang Y. Isogeometric fluid-structure interaction: theory, algorithms, and computations. *Computational Mechanics* 2008; **43**:3–37.
4. Bazilevs Y, Calo VM, Zhang Y, Hughes TJR. Isogeometric fluid-structure interaction analysis with applications to arterial blood flow. *Computational Mechanics* 2006; **38**:310–322.
5. Zhang Y, Bazilevs Y, Goswami S, Bajaj CL, Hughes TJR. Patient-specific vascular NURBS modeling for isogeometric analysis of blood flow. *Computer Methods in Applied Mechanics and Engineering* 2007; **196**: 2943–2959.
6. Bueno J, Bona-Casas C, Bazilevs Y, Gomez H. Interaction of complex fluids and solids: theory, algorithms and application to phase-change-driven implosion. *Computational Mechanics* 2015; **55**(6):1105–1118.
7. Kamensky D, Hsu MC, Schillinger D, Evans JA, Aggarwal A, Bazilevs Y, Sacks MS, Hughes TJR. An immersogeometric variational framework for fluid-structure interaction: application to bioprosthetic heart valves. *Computer Methods in Applied Mechanics and Engineering* 2015; **284**:1005–1053.

DOI: 10.1002/nme

8. Casquero H, Bona-Casas C, Gomez H. A NURBS-based immersed methodology for fluid-structure interaction. *Computer Methods in Applied Mechanics and Engineering* 2015; **284**:943–970.
9. Hsu MC, Kamensky D, Bazilevs Y, Sacks M, Hughes TJR. Fluid-structure interaction analysis of bioprosthetic heart valves: significance of arterial wall deformation. *Computational Mechanics* 2014; **54**:1055–1071 (English).
10. Bazilevs Y, Calo VM, Cottrell JA, Hughes TJR, Reali A, Scovazzi G. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Computer Methods in Applied Mechanics and Engineering* 2007; **197**:173–201.
11. Evans JA, Hughes TJR. Isogeometric divergence-conforming B-splines for the unsteady Navier–Stokes equations. *Journal of Computational Physics* 2013; **241**:141–167.
12. Gomez H, Cueto-Felgueroso L, Juanes R. Three-dimensional simulation of unstable gravity-driven infiltration of water into a porous medium. *Journal of Computational Physics* 2013; **238**:217–239.
13. Gomez H, Hughes TJR, Nogueira X, Calo VM. Isogeometric analysis of the isothermal Navier–Stokes–Korteweg equations. *Computer Methods in Applied Mechanics and Engineering* 2010; **199**:1828–1840.
14. Liu Ju, Gomez H, Evans JA, Hughes TJR, Landis CM. Functional entropy variables: a new methodology for deriving thermodynamically consistent algorithms for complex fluids, with particular reference to the isothermal Navier–Stokes–Korteweg equations. *Journal of Computational Physics* 2013; **248**:47–86.
15. Gomez H, Nogueira X. A new space–time discretization for the Swift–Hohenberg equation that strictly respects the Lyapunov functional. *Communications in Nonlinear Science and Numerical Simulation* 2012; **17**:4930–4946.
16. Gomez H, Calo VM, Bazilevs Y, Hughes TJR. Isogeometric analysis of the Cahn–Hilliard phase-field model. *Computacional Methods in Applied Mechanics and Engineering* 2008; **197**:4333–4352.
17. Gomez H, Hughes TJR. Provably unconditionally stable, second-order time-accurate, mixed variational methods for phase-field models. *Journal of Computational Physics* 2011; **230**:5310–5327.
18. Gomez H, Nogueira X. An unconditionally energy-stable method for the phase field crystal equation. *Computer Methods in Applied Mechanics and Engineering* 2012; **249**:52–61.
19. Thiele U, Archer AJ, Robbins MJ, Gomez H, Knobloch E. Localized states in the conserved Swift–Hohenberg equation with cubic nonlinearity. *Physical Review E* 2013; **87**(4):042915.
20. Gomez H, París J. Numerical simulation of asymptotic states of the damped Kuramoto–Sivashinsky equation. *Physical Review E* 2011; **83**:046702.
21. Vilanova G, Colominas I, Gomez H. Capillary networks in tumor angiogenesis: from discrete endothelial cells to phase-field averaged descriptions via isogeometric analysis. *International Journal for Numerical Methods in Biomedical Engineering* 2013; **29**:1015–1037.
22. Vilanova G, Colominas I, Gomez H. Coupling of discrete random walks and continuous modeling for three-dimensional tumor-induced angiogenesis. *Computational Mechanics* 2013; **53**:449–464.
23. Cottrell JA, Reali A, Bazilevs Y, Hughes TJR. Isogeometric analysis of structural vibrations. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:5257–5296.
24. Hughes TJR, Reali A, Sangalli G. Duality and unified analysis of discrete approximations in structural dynamics and wave propagation: comparison of p-method finite elements with k-method NURBS. *Computer Methods in Applied Mechanics and Engineering* 2008; **197**:4104–4124.
25. Lipton S, Evans JA, Bazilevs Y, Elguedj T, Hughes TJR. Robustness of isogeometric structural discretizations under severe mesh distortion. *Computer Methods in Applied Mechanics and Engineering* 2010; **199**:357–373.
26. Elguedj T, Bazilevs Y, Calo VM, Hughes TJRH. $\overline{B}$ and $\overline{F}$ projection methods for nearly incompressible linear and non-linear elasticity and plasticity using higher-order NURBS elements. *Computer Methods in Applied Mechanics and Engineering* 2008; **197**:2732–2762.
27. Dhote RP, Gomez H, Melnik RNV, Zu J. Isogeometric analysis of a dynamic thermo-mechanical phase-field model applied to shape memory alloys. *Computational Mechanics* 2013; **53**:1235–1250.
28. Dhote RP, Gomez H, Melnik RNV, Zu J. Shape memory alloy nanostructures with coupled dynamic thermo-mechanical effects. *Computer Physics Communications* 2015; **192**:48–53.
29. Dhote RP, Gomez H, Melnik RNV, Zu J. 3D coupled thermo-mechanical phase-field modeling of shape memory alloy dynamics via isogeometric analysis. *Computers & Structures* 2015; **154**:48–58.
30. Kiendl J, Bletzinger KU, Linhard J, Wuchner R. Isogeometric shell analysis with Kirchhoff–Love elements. *Computer Methods in Applied Mechanics and Engineering* 2009; **198**:3902–3914.
31. Kiendl J, Bazilevs Y, Hsu MC, Wüchner R, Bletzinger KU. The bending strip method for isogeometric analysis of Kirchhoff–Love shell structures comprised of multiple patches. *Computer Methods in Applied Mechanics and Engineering* 2010; **199**:2403–2416.
32. Nguyen-Thanh N, Kiendl J, Nguyen-Xuan H, Wuchner R, Bletzinger KU, Bazilevs Y, Rabczuk T. Rotation free isogeometric thin shell analysis using PHT-splines. *Computer Methods in Applied Mechanics and Engineering* 2011; **200**:3410–3424.
33. Kiendl J, Hsu MC, Wu MCH, Reali A. Isogeometric Kirchhoff–Love shell formulations for general hyperelastic materials. *Computer Methods in Applied Mechanics and Engineering* 2015; **291**:280–303.
34. De Lorenzis L, Temizer I, Wriggers P, Zavarise G. A large deformation frictional contact formulation using NURBS-based isogeometric analysis. *International Journal for Numerical Methods in Engineering* 2011; **87**:1278–1300.
35. Dimitri R, De Lorenzis L, Scott MA, Wriggers P, Taylor RL, Zavarise G. Isogeometric large deformation frictionless contact using T-splines. *Computer Methods in Applied Mechanics and Engineering* 2014; **269**:394–414.
36. Dimitri R, De Lorenzis L, Wriggers P, Zavarise G. NURBS-and T-spline-based isogeometric cohesive zone modeling of interface debonding. *Computational Mechanics* 2014; **54**:369–388.

DOI: 10.1002/nme

37. Sederber TW, Zheng J, Bakenov A, Nasri A. T-splines and T-NURCCs. *ACM Transactions on Graphics* 2003; **22**:477–484.
38. Zhang Y, Wang W, Hughes TJR. Solid T-spline construction from boundary representations for genus-zero geometry. *Computer Methods in Applied Mechanics and Engineering* 2012; **249-252**:185–197.
39. Zhang Y, Wang W, Hughes TJRH. Conformal solid T-spline construction from boundary T-spline representations. *Computational Mechanics* 2013; **51**:1051–1059.
40. Wang W, Zhang Y, Liu L, Hughes TJR. Trivariate solid T-spline construction from boundary triangulations with arbitrary genus topology. *Computer-Aided Design* 2013; **45**(2):351–360.
41. Liu L, Zhang Y, Hughes TJR, Scott MA, Sederberg TW. Volumetric T-spline construction using boolean operations. *Engineering with Computers* 2014; **30**:425–439.
42. Liu L, Zhang Y, Liu Y, Wang W. Feature-preserving t-mesh construction using skeleton-based polycubes. *Computer-Aided Design* 2015; **58**:162–172.
43. Bazilevs Y, Calo VM, Cottrell JA, Evans JA, Hughes TJR, Lipton S, Scott MA, Sederberg TW. Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering* 2010; **199**:229–263.
44. Li X, Zheng J, Sederberg TW, Hughes TJR, Scott MA. On linear independence of T-spline blending functions. *Computer Aided Geometric Design* 2012; **29**:63–76.
45. Scott MA, Simpson RN, Evans JA, Lipton S, Bordas SPA, Hughes TJR, Sederberg TW. Isogeometric boundary element analysis using unstructured T-splines. *Computer Methods in Applied Mechanics and Engineering* 2013; **254**:197–221.
46. Scott MA, Borden MJ, Verhoosel CV, Sederberg TW, Hughes TJR. Isogeometric finite element data structures based on Bézier extraction of T-splines. *International Journal for Numerical Methods in Engineering* 2011; **88**:126–156.
47. Borden MJ, Scott MA, Evans JA, Hughes TJR. Isogeometric finite element data structures based on Bézier extraction of NURBS. *International Journal for Numerical Methods in Engineering* 2011; **87**(1-5):15–47.
48. Scott MA, Li X, Sederberg TW, Hughes TJR. Local refinement of analysis-suitable T-splines. *Computer Methods in Applied Mechanics and Engineering* 2012; **213-216**:206–222.
49. Wang W, Zhang Y, Scott MA, Hughes TJR. Converting an unstructured quadrilateral mesh to a standard T-spline surface. *Computational Mechanics* 2011; **48**:477–498.
50. Liu L, Zhang YJ, Wei X. Weighted T-splines with application in reparameterizing trimmed NURBS surfaces. *Computer Methods in Applied Mechanics and Engineering* 2015; **295**:108–126.
51. Evans EJ, Scott MA, Li X, Thomas DC. Hierarchical T-splines: analysis-suitability, Bézier extraction, and application as an adaptive basis for isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering* 2015; **284**:1–20.
52. Auricchio F, Da Veiga LB, Hughes TJR, Reali A, Sangalli G. Isogeometric collocation methods. *Mathematical Models and Methods in Applied Sciences* 2010; **20**:2075–2107.
53. Gomez H, Reali A, Sangalli G. Accurate, efficient, and (iso)geometrically flexible collocation methods for phase-field models. *Journal of Computational Physics* 2014; **262**:153–171.
54. Schillinger D, Evans JA, Reali A, Scott MA, Hughes TJR. Isogeometric collocation: cost comparison with Galerkin methods and extension to adaptive hierarchical NURBS discretizations. *Computer Methods in Applied Mechanics and Engineering* 2013; **267**:170–232.
55. Reali A, Gomez H. An isogeometric collocation approach for Bernoulli–Euler beams and Kirchhoff plates. *Computer Methods in Applied Mechanics and Engineering* 2015; **284**:623–636.
56. Casquero H, Lei L, Zhang J, Reali A, Gomez H. *Isogeometric collocation using analysis-suitable T-splines of arbitrary degree*, 2015. submitted for publication.
57. Borden MJ, Verhoosel CV, Scott MA, Hughes TJR, Landis CM. A phase-field description of dynamic brittle fracture. *Computer Methods in Applied Mechanics and Engineering* 2012; **217-220**:77–95.
58. Verhoosel CV, Scott MA, de Borst R, Hughes TJR. An isogeometric approach to cohesive zone modeling. *International Journal for Numerical Methods in Engineering* 2011; **87**:336–360.
59. Verhoosel CV, Scott MA, Hughes TJR, de Borst R. An isogeometric analysis approach to gradient damage models. *International Journal for Numerical Methods in Engineering* 2011; **86**:115–134.
60. Dimitri R, Lorenzis LD, Scott MA, Wriggers P, Taylor RL, Zavarise G. Isogeometric large deformation frictionless contact using T-splines. *Computer Methods in Applied Mechanics and Engineering* 2014; **269**:394–414.
61. Simpson RN, Scott MA, Taus M, Thomas DC, Lian H. Acoustic isogeometric boundary element analysis. *Computer Methods in Applied Mechanics and Engineering* 2014; **269**:265–290.
62. Bazilevs Y, Hsu MC, Scott MA. Isogeometric fluid-structure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines. *Computer Methods in Applied Mechanics and Engineering* 2012; **249-252**:28–41.
63. Hughes TJR, Liu WK, Zimmermann TK. Lagrangian–Eulerian finite element formulation for incompressible viscous flows. *Computer methods in applied mechanics and engineering* 1981; **29**:329–349.
64. Bazilevs Y, Hsu MC, Zhang Y, Wang W, Kvamsdal T, Hentschel S, Isaksen JG. Computational vascular fluid-structure interaction: methodology and application to cerebral aneurysms. *Biomechanics and Modeling in Mechanobiology* 2010; **9**:481–498.
65. Bazilevs Y, Gohean JR, Hughes TJR, Moser RD, Zhang Y. Patient-specific isogeometric fluid-structure interaction analysis of thoracic aortic blood flow due to implantation of the Jarvik 2000 left ventricular assist device. *Computer Methods in Applied Mechanics and Engineering* 2009; **198**:3534–3550.

DOI: 10.1002/nme

66. Isaksen JG, Bazilevs Yuri, Kvamsdal T, Zhang Y, Kaspersen JH, Waterloo K, Romner B, Ingebrigtsen T. Determination of wall tension in cerebral artery aneurysms by numerical simulation. *Stroke* 2008; **39**:3172–3178.
67. Bazilevs Y, Hsu MC, Zhang Y, Wang W, Liang X, Kvamsdal T, Brekken R, Isaksen JG. A fully-coupled fluid-structure interaction simulation of cerebral aneurysms. *Computational Mechanics* 2010; **46**:3–16.
68. Zhang Y, Wang W, Liang X, Bazilevs Y, Hsu MC, Kvamsdal T, Brekken R, Isaksen J. High-fidelity tetrahedral mesh generation from medical imaging data for fluid-structure interaction analysis of cerebral aneurysms. *Computer Modeling in Engineering and Sciences (CMES)* 2009; **42**:131–150.
69. Bazilevs Y, Hsu MC, Akkerman I, Wright S, Takizawa K, Henicke B, Spielman T, Tezduyar TE. 3D simulation of wind turbine rotors at full scale. Part I: geometry modeling and aerodynamics. *International Journal for Numerical Methods in Fluids* 2011; **65**:207–235.
70. Bazilevs Y, Hsu MC, Kiendl J, Wuchner R, Bletzinger KU. 3D simulation of wind turbine rotors at full scale. Part II: fluid-structure interaction modeling with composite blades. *International Journal for Numerical Methods in Fluids* 2011; **65**:236–253.
71. Farhat C, van der Zee KG, Geuzaine P. Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity. *Computer methods in applied mechanics and engineering* 2006; **195**:1973–2001.
72. Tezduyar TE, Takizawa K, Moorman C, Wright S, Christopher J. Space-time finite element computation of complex fluid-structure interactions. *International Journal for Numerical Methods in Fluids* 2010; **64**:1201–1218.
73. van Brummelen EH, van der Zee KG, de Borst R. Space/time multigrid for a fluid-structure-interaction problem. *Applied Numerical Mathematics* 2008; **58**:1951–1971.
74. Bazilevs Y, Takizawa K, Tezduyar TE. *Computational Fluid-structure Interaction: Methods and Applications*. John Wiley & Sons, 2012.
75. Peskin CS. Flow patterns around heart valves: a numerical method. *Journal of Computational Physics* 1972; **10**: 252–271.
76. McQueen DM, Peskin CS. Computer-assisted design of butterfly bileaflet valves for the mitral position. *Scandinavian Journal of Thoracic and Cardiovascular Surgery* 1985; **19**:139–148.
77. Beyer Jr. RP. A computational model of the cochlea using the immersed boundary method. *Journal of Computational Physics* 1992; **98**:145–162.
78. Dillon R, Fauci L, Fogelson A, Gaver III D. Modeling biofilm processes using the immersed boundary method. *Journal of Computational Physics* 1996; **129**:57–73.
79. Gil AJ, Carreño AA, Bonet J, Hassan O. The immersed structural potential method for haemodynamic applications. *Journal of Computational Physics* 2010; **229**(22):8613–8641.
80. Mountrakis L, Lorenz E, Hoekstra AG. Where do the platelets go? A simulation study of fully resolved blood flow through aneurysmal vessels. *Interface focus* 2013; **3**(2):20120089.
81. Mountrakis L, Lorenz E, Malaspinas O, Alowayyed S, Chopard B, Hoekstra AG. Parallel performance of an IB-LBM suspension simulation framework. *Journal of Computational Science* 2015; **9**:45–50.
82. Zhang L, Gerstenberger A, Wang X, Liu WK. Immersed finite element method. *Computer Methods in Applied Mechanics and Engineering* 2004; **193**:2051–2067.
83. Liu WK, Liu Y, Farrell D, Zhang L, Wang XS, Fukui Y, Patankar N, Zhang Y, Bajaj C, Lee J, Hong J, Chen X, Hsu H. Immersed finite element method and its applications to biological systems. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:1722–1749.
84. Gay M, Zhang L, Liu WK. Stent modeling using immersed finite element method. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:4358–4370.
85. Chung J, Hulbert GM. A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-Îś method. *Journal of Applied Mechanics* 1993; **60**:371–375.
86. Jansen KE, Whiting CH, Hulbert GM. Generalized-α method for integrating the filtered Navier–Stokes equations with a stabilized finite element method. *Computer Methods in Applied Mechanics and Engineering* 2000; **190**: 305–319.
87. Beirao da Veiga L, Buffa A, Sangalli G, VÃązquez R. Analysis suitable T-splines of arbitrary degree: definition, linear independence, and approximation properties. *Mathematical Models and Methods in Applied Sciences* 2013; **23**:1979–2003.
88. Beirao da Veiga L, Buffa A, Cho D, Sangalli G. Analysis-suitable T-splines are dual-compatible. *Computer Methods in Applied Mechanics and Engineering* 2012; **249252**(0):42–51. Higher Order Finite Element and Isogeometric Methods.
89. Zienkiewicz OC, Taylor RL, Zienkiewicz OC, Taylor RL. *The Finite Element Method*. McGraw-hill London, 1977.
90. Hughes TJR. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis 2000;* Cited By (since 1996)1023.
91. De Boor C. A practical guide to splines. *Mathematics of Computation* 1978.
92. Marsden JE, Hughes TJR. *Mathematical Foundations of Elasticity*. Courier Dover Publications: Springer-Verlag New York, 1994.
93. Hesch C, Gil AJ, Arranz CA, Bonet J. On continuum immersed strategies for fluid-structure interaction. *Computer Methods in Applied Mechanics and Engineering* 2012; **247-248**:51–64.
94. Simo JC, Hughes TJR. *Computational Inelasticity*. Springer: New York, 2008.

DOI: 10.1002/nme

95. Hesch C, Gil AJ, Carreño AA, Bonet J, Betsch P. A mortar approach for fluid-structure interaction problems: immersed strategies for deformable and rigid bodies. *Computer Methods in Applied Mechanics and Engineering* 2014; **278**:853–882.
96. Balay S, Adams MF, Brown J, Brune P, Buschelman K, Eijkhout V, Gropp WD, Kaushik D, Knepley MG, McInnes LC, Rupp K, Smith BF, Zhang H. *PETSc Web page*, 2014. (Available from: http://www.mcs.anl.gov/petsc).
97. Boffi D, Cavallini N, Gastaldi L. The finite element immersed boundary method with distributed Lagrange multiplier. *arXiv preprint arXiv:1407.5184* 2014.
98. Auricchio F, Boffi D, Gastaldi L, Lefieux A, Reali A. On a fictitious domain method with distributed Lagrange multiplier for interface problems. *Applied Numerical Mathematics* 2014; **95**:36–50.
99. Auricchio F, Boffi D, Gastaldi L, Lefieux A, Reali A. A study on unfitted 1d finite element methods. *Computers & Mathematics with Applications* 2014; **68**(12):2080–2102.

DOI: 10.1002/nme

D Paper #4: "NURBS-based numerical proxies for red blood cells and circulating tumor cells in microscale blood flow"

NURBS-based numerical proxies for red blood cells and circulating tumor cells in microscale blood flow

Hugo Casquero[a,*], Carles Bona-Casas[b], Hector Gomez[a]

[a]*Departamento de Métodos Matemáticos, Universidade da Coruña, Campus de A Coruña, 15071, A Coruña, Spain.*
[b]*Departament de Física, Universitat de les Illes Balears, 07122, Palma de Mallorca, Spain*

Abstract

We explore the use of NURBS-based immersed fluid-structure interaction algorithms to model the dynamics and rheology of red blood cells (RBCs) and nucleated cells. Prime examples of cells with nucleus that are relevant to this study are white blood cells (WBCs) and circulating tumor cells (CTCs). In this work, RBCs are modeled as thin solid membranes called *capsules*. To model cells with nucleus we introduce the concept of *compound capsule* which explicitly takes into account the nucleus as a bulky deformable solid. Our results indicate that to reproduce the behavior of RBCs in shear and parabolic flows, it is crucial to accurately solve the mass conservation equation near the fluid-solid interface. We show results of hyperelastic capsules and compound capsules in two- and three-dimensional settings. Finally, taking advantage of the geometric flexibility of our method, we simulate how a CTC passes through a narrowing. This is a feature of CTCs that has recently triggered excitement in the experimental community.

Keywords:
Isogeometric analysis, Immersed fluid-structure interaction, Hyperelastic compound capsules, Tank-treading motion, Tumbling motion, Fåhræus-Lindqvist Effect

1. Introduction

Blood is a complex fluid that is primarily made of blood plasma and red blood cells (RBCs). The concentration of RBCs (hematocrit) in a healthy human body is in average 45% of the blood volume. The remaining formed elements of blood such as, e.g., white blood cells (WBCs) and platelets, add up to less than 1% of the total blood volume. The mechanical behavior of blood depends strongly on the scale of study. In large vessels with a diameter greater than 500 µm, considering blood as a Newtonian fluid with a constant viscosity is sufficiently accurate for most applications. However, in vessels with smaller diameters, blood behaves as a non-Newtonian fluid. In fact, to be able to match experimental

*Corresponding author.
Email address: hugo.casquero@udc.es (Hugo Casquero)

results of blood flow in cylindrical tubes with Poiseuille's law, it is necessary to define an *apparent viscosity* that depends at least on the tube's diameter and the hematocrit. This phenomenon is known as the Fåhræus-Lindqvist effect [1].

In most cases, RBCs are the blood constituent principally responsible for the dynamics and rheology of blood, which is related with the fact that RBCs are more numerous and more deformable than the remaining formed elements of blood. One way to study microscale blood flow is to explicitly model RBCs. This is the approach chosen in the present work and it leads to a fluid-structure interaction (FSI) problem that we solve using an immersed approach. A RBC is composed by a lipid bilayer membrane with an attached spectrin network called cytoskeleton which encapsulates a concentrated hemoglobin solution that behaves as a Newtonian incompressible fluid. Under physiological conditions, both the RBC volume and external area are nearly constant. In a quiescent fluid, healthy RBCs have a biconcave shape. Nevertheless, under nonequilibrium conditions such us, e.g., pure shear flow and parabolic flow, RBCs exhibit a variety of interesting shapes and motions [2, 3]. In the last decades, the reproduction of experimental shapes and motions has been the most widespread benchmark test used in order to evaluate the accuracy and efficiency of different RBC models in blood flow*. The two most common continuum-based approaches for modeling RBCs are the so-called *vesicles* [5, 6, 7] and *capsules* [8]. Vesicles consider the bending resistance of the lipid bilayer membrane which can be directly controlled through a parameter called bending rigidity, but they neglect the shear resistance of the cytoskeleton. Capsules consider both bending and shear resistances, but both types of resistances are usually controlled through the same parameter, namely, the shear modulus. In this work, we use capsules as our model for RBCs.

Blood plasma behaves as a Newtonian incompressible fluid whose average density and dynamic viscosity are $1.025\,\mathrm{g/cm^3}$ and $1.2\,\mathrm{mPa\ s}$, respectively [9]. Due to the low values of the Reynolds number usually encountered in microcirculation problems, it is quite common to assume Stokes flow to model blood plasma at small scales. However, recent findings show that inertial effects do play an important role in microcirculation, especially when the flow domain is curved [10, 11]. These observations are giving rise to an entirely new field call *inertial microfluidics* and are expected to have practical applications in the next few years. Motivated by this experimental evidence, we do not assume Stokes flow, but solve the Navier-Stokes equations using the variational multiscale (VMS) technique [12, 13, 14].

In the last few years, the continuum-based modeling of WBCs has received increasing attention. The goal is to elucidate a plethora of open questions about flows that involve both RBCs and WBCs. Besides of a membrane, a WBC contains a nucleus, organelles, and fibers, which are suspended in a viscous cytosol [15]. Therefore, modeling WBCs either as bulky bodies or using the same models used for RBCs (e.g., vesicles or capsules) is not accurate in a number of situations. In [16], a model for WBCs is proposed under the assumption that the behavior of a WBC is dominated by its nucleus, its viscous cytosol,

*For a thorough discussion of the different types of models used for RBCs, the interested reader is remitted to [4].

and its membrane, that is, the organelles and other internal constituents are neglected. Ref. [16] introduces the concept of *compound vesicles* in which the membrane is modeled as a vesicle (i.e., they only consider bending resistance) and the nucleus is modeled as a rigid body (i.e., they neglect its strains). The modeling of other types of nucleated cells, e.g., circulating tumor cells (CTCs) is also an appealing subject. CTCs are cancerous cells that escape from the tumor, enter the circulatory system through a process called intravasation and acquire the ability to survive in this new environment. Eventually, some of these cells may extravasate, that is, leave the circulatory system, settle in a new organ and possibly create a secondary tumor. This process is part of the so-called metastatic cascade. We believe that understanding the behavior of CTCs in microcirculation would lead to a deeper understanding of the metastatic cascade which could help to develop therapies to stop cancer spreading [17]. It may also permit to design devices that allow to isolate CTCs from a blood sample with diagnostic purposes.

In this work, we use a NURBS-based immersed FSI approach to study the behavior of capsules in shear and parabolic flows. Our numerical results are compared with experimental results for RBCs. Good agreement is found as long as the incompressibility constraint is adequately satisfied near the fluid-solid interface. Ensuring that the incompressibility constraint is satisfied is quite challenging due to the assumption of continuous pressure across the fluid-solid interface which is done in all immersed methods that we are aware of. Moreover, handling the large strains undergone by the numerical proxies of RBCs is one of the main bottlenecks of mesh-based immersed FSI approaches. In [18, 19], the authors claim that piecewise linear Lagrange polynomials cannot deal with the large strains found in their simulations and they propose to use the meshfree reproducing kernel particle method for representing the capsules instead [20, 21]. In our case, the higher inter-element continuity of NURBS gives us sufficient robustness [22] to handle the large strains undergone by the capsules in all of our simulations. In this paper, we introduce the concept of *compound capsules* to model nucleated cells. A compound capsule is formed by a capsule, an incompressible Newtonian fluid enclosed by the capsule, and a bulky deformable solid suspended in the fluid, which represent the membrane, the cytosol, and the nucleus of the cell, respectively. The dynamics of compound capsules in shear flow are analyzed in two- and three-dimensional settings. Our results are in agreement with those of compound vesicles. Additionally, the geometric flexibility of our immersed approach enables us to solve problems on non-trivial geometries.

The paper is organized as follows: Section 2 describes our FSI modeling framework and emphasizes the main challenges faced when an immersed approach is used. Section 3 studies the dynamics of capsules and compound capsules in shear flows. In Section 4, we focus on the behavior of RBCs and CTCs in parabolic flows. Section 5 presents our conclusions and future directions of work.

2. FSI simulation using an immersed approach

Our FSI formulation at the continuous and discrete levels was thoroughly derived in our two earlier works [23, 24]. In this Section, we summarize our hybrid variational-collocation

immersed method for FSI and focus on a number of important subjects, namely, how to improve local mass conservation close to the fluid-solid interface, how to alleviate the issue of suboptimal quadrature rules that arises in immersed FSI through higher inter-element continuity, the coupling strategy, and the nonlinear and linear solvers used for the final system of algebraic equations.

2.1. Governing equations

Let $d = \{2, 3\}$ be the number of spatial dimensions. Let $\Omega_t^1 \subset \mathbb{R}^d$ and $\Omega_t^2 \subset \mathbb{R}^d$ be two open sets that represent the time-dependent domains occupied by a viscous incompressible fluid and a hyperelastic incompressible solid, respectively. Γ_t^1 and Γ_t^2 are their corresponding boundaries. These two regions meet at the fluid-solid interface $\Gamma_t^I = \overline{\Omega_t^1} \cap \overline{\Omega_t^2}$. Let $\Omega \subset \mathbb{R}^d$ be an open set that represents the domain occupied by both the fluid and the solid, that is, $\overline{\Omega} = \overline{\Omega_t^1 \cup \Omega_t^2}$. In all the simulations presented in this paper, Ω is time independent. Therefore, we will assume that Ω is fixed in time to simplify the description of our algorithm. The boundary of Ω is denoted by Γ. Let $\boldsymbol{v} : \Omega \times (0,T) \mapsto \mathbb{R}^d$, $p : \Omega \times (0,T) \mapsto \mathbb{R}$, and $\boldsymbol{u}_2 : \Omega_0^2 \times (0,T) \mapsto \mathbb{R}^d$ denote the Eulerian velocity, the Eulerian pressure, and the Lagrangian displacement, respectively, where $(0,T)$ is the time interval of interest and Ω_0^2 is a reference configuration of Ω_t^2. Let $\boldsymbol{X} \in \Omega_0^2$, $\boldsymbol{x} \in \Omega$, and $\boldsymbol{\varphi} : \Omega_0^2 \times (0,T) \mapsto \Omega_t^2$ be a material point, a spatial point, and the deformation mapping $\boldsymbol{\varphi}(\boldsymbol{X}, t) = \boldsymbol{X} + \boldsymbol{u}_2(\boldsymbol{X}, t)$, respectively.

The three equations that define our problem are the linear momentum balance equation, the mass conservation equation, and the kinematic relation between the Eulerian velocity and the Lagrangian displacement, i.e.,

$$\rho_1 \left(\frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v} \cdot \nabla_{\boldsymbol{x}} \boldsymbol{v} \right) = \nabla_{\boldsymbol{x}} \cdot \boldsymbol{\sigma}_1 + \rho_1 \boldsymbol{g} + \boldsymbol{\mathcal{F}} \qquad \text{in} \quad \Omega \times (0,T) \ , \tag{1}$$

$$\nabla_{\boldsymbol{x}} \cdot \boldsymbol{v} = 0 \qquad \text{in} \quad \Omega \times (0,T) \ , \tag{2}$$

$$\frac{\partial \boldsymbol{u}_2}{\partial t} = \boldsymbol{v} \qquad \text{in} \quad \Omega_t^2 \times (0,T) \ , \tag{3}$$

with

$$\boldsymbol{\mathcal{F}} = \begin{cases} 0, & \boldsymbol{x} \in \Omega_t^1 \\ (\rho_1 - \rho_2) \left(\dfrac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v} \cdot \nabla_{\boldsymbol{x}} \boldsymbol{v} - \boldsymbol{g} \right) + \nabla_{\boldsymbol{x}} \cdot (\boldsymbol{\sigma}_2 - \boldsymbol{\sigma}_1) \, , & \boldsymbol{x} \in \Omega_t^2 \end{cases} \ , \tag{4}$$

and constitutive laws

$$\boldsymbol{\sigma}_1 = -p\boldsymbol{I} + 2\mu \nabla_{\boldsymbol{x}}^{\text{sym}} \boldsymbol{v} \ , \tag{5}$$

$$\boldsymbol{\sigma}_2 = -p\boldsymbol{I} + \boldsymbol{F}\boldsymbol{S}\boldsymbol{F}^T / J \ . \tag{6}$$

Here, ρ_1 and ρ_2 are the fluid and solid densities, respectively, $\boldsymbol{\sigma}_1$ and $\boldsymbol{\sigma}_2$ are the fluid and solid Cauchy stress tensors, respectively, $\boldsymbol{g}$ is an external body force acting on the system, $\boldsymbol{I}$ denotes the identity tensor in $\mathbb{R}^{d \times d}$, μ is the dynamic viscosity of the fluid, $\nabla_{\boldsymbol{x}}^{\text{sym}}(\cdot)$ is the symmetric gradient operator given by $\nabla_{\boldsymbol{x}}^{\text{sym}} \boldsymbol{v} = (\nabla_{\boldsymbol{x}} \boldsymbol{v} + \nabla_{\boldsymbol{x}} \boldsymbol{v}^T)/2$, $\boldsymbol{F} : \Omega_0^2 \times (0,T) \mapsto \mathbb{R}^{d \times d}$ is the deformation gradient given by $\boldsymbol{F} = \nabla_{\boldsymbol{X}} \boldsymbol{\varphi} = \boldsymbol{I} + \nabla_{\boldsymbol{X}} \boldsymbol{u}_2$, $\boldsymbol{S}$ denotes the second

Piola-Kirchhoff stress tensor of the considered hyperelastic incompressible solid, and $J = \det(\boldsymbol{F})$ is the Jacobian determinant. At the continuous level, $J = 1$ because the solid is incompressible.

In order to achieve a well-posed FSI problem, we need to impose suitable initial and boundary conditions. Regarding initial conditions, the Eulerian velocity and Lagrangian displacement need to be given at the initial time. To simplify the definition of boundary conditions, we consider that the solid is fully immersed in the fluid, which is the case of interest for RBCs and CTCs. In this scenario, the boundary of the solid coincides with the fluid-solid interface ($\Gamma_t^2 = \Gamma_t^I$) and we impose two boundary conditions at the interface, namely, the no-slip boundary condition and the stress compatibility between the fluid and solid Cauchy stress tensors. The no-slip boundary condition is imposed through Eq. (3) and the stress compatibility condition will be naturally enforced by the variational form of (1). On the boundary of the fluid $\Gamma_t^1 = \Gamma$, we assume the standard split in Dirichlet boundary conditions on Γ_D and Neumann boundary conditions on Γ_N. The sets Γ_D and Γ_N satisfy the relations $\Gamma = \Gamma_D \cup \Gamma_N$ and $\Gamma_D \cap \Gamma_N = \emptyset$.

Eq. (3) will be discretized in space using isogeometric collocation [25, 26, 27, 28], therefore, it is not necessary to recast it in weak form. However, Eqs. (1) and (2) will be discretized in space using a weighted residual formulation, viz., the VMS approach proposed in [14]. The basic idea behind the VMS technique is to split the velocity and the pressure into two components, namely, $\boldsymbol{v} = \widetilde{\boldsymbol{v}} + \boldsymbol{v}'$ and $p = \widetilde{p} + p'$. Here, $\widetilde{\boldsymbol{v}}$ and $\widetilde{p}$ are finite-dimensional coarse-scale components while $\boldsymbol{v}'$ and p' are infinite-dimensional fine-scale components. Then, we multiply Eqs. (1) and (2) by weight functions $\widetilde{\boldsymbol{w}}$ and $\widetilde{q}$, respectively, integrate over Ω, and perform various mathematical operations described in detail in [14]. Following the aforementioned steps, we derive the following weak formulation of Eqs. (1) and (2):

$$B^{ED}\left(\{\widetilde{\boldsymbol{w}}, \widetilde{q}\}, \{\widetilde{\boldsymbol{v}} + \boldsymbol{v}', \widetilde{p} + p'\}\right) - L^{ED}\left(\widetilde{\boldsymbol{w}}\right) + B^{LD}\left(\widetilde{\boldsymbol{w}}, \widetilde{\boldsymbol{v}}; \boldsymbol{u}_2\right) = 0 \ . \tag{7}$$

with

$$\begin{aligned} B^{ED}\left(\{\widetilde{\boldsymbol{w}}, \widetilde{q}\}, \{\widetilde{\boldsymbol{v}} + \boldsymbol{v}', \widetilde{p} + p'\}\right) = & \left(\widetilde{\boldsymbol{w}}, \frac{\partial \widetilde{\boldsymbol{v}}}{\partial t} + \widetilde{\boldsymbol{v}} \cdot \nabla_{\boldsymbol{x}} \widetilde{\boldsymbol{v}}\right)_\Omega \\ & - \left(\nabla_{\boldsymbol{x}} \cdot \widetilde{\boldsymbol{w}}, \widetilde{p}\right)_\Omega + \left(\nabla_{\boldsymbol{x}}^{\mathrm{sym}} \widetilde{\boldsymbol{w}}, 2\nu \nabla_{\boldsymbol{x}}^{\mathrm{sym}} \widetilde{\boldsymbol{v}}\right)_\Omega \\ & + \left(\widetilde{q}, \nabla_{\boldsymbol{x}} \cdot \widetilde{\boldsymbol{v}}\right)_\Omega + \left(\widetilde{\boldsymbol{w}}, \boldsymbol{v}' \cdot \nabla \widetilde{\boldsymbol{v}}\right)_\Omega \\ & - \left(\nabla_{\boldsymbol{x}} \widetilde{\boldsymbol{w}}, (\widetilde{\boldsymbol{v}} + \boldsymbol{v}') \otimes \boldsymbol{v}'\right)_\Omega - \left(\nabla_{\boldsymbol{x}} \cdot \widetilde{\boldsymbol{w}}, p'\right)_\Omega \\ & - \left(\nabla_{\boldsymbol{x}} \widetilde{q}, \boldsymbol{v}'\right)_\Omega \ , \end{aligned} \tag{8}$$

$$L^{ED}\left(\widetilde{\boldsymbol{w}}\right) = \left(\widetilde{\boldsymbol{w}}, \boldsymbol{g}\right)_\Omega + \left(\widetilde{\boldsymbol{w}}, \boldsymbol{h}/\rho_1\right)_{\Gamma_N} \ , \tag{9}$$

$$
\begin{aligned}
B^{LD}\left(\widetilde{\boldsymbol{w}},\widetilde{\boldsymbol{v}};\boldsymbol{u}_2\right) = &-\left(\widetilde{\boldsymbol{w}},\left(1-\frac{\rho_2}{\rho_1}\right)\frac{\partial\widetilde{\boldsymbol{v}}}{\partial t}\right)_{\Omega_0^2}\\
&-\left(\widetilde{\boldsymbol{w}},\left(1-\frac{\rho_2}{\rho_1}\right)\widetilde{\boldsymbol{v}}\cdot\nabla_{\boldsymbol{x}}\widetilde{\boldsymbol{v}}\right)_{\Omega_0^2}\\
&-\left(\nabla_{\boldsymbol{x}}^{\mathrm{sym}}\widetilde{\boldsymbol{w}},2\nu\nabla_{\boldsymbol{x}}^{\mathrm{sym}}\widetilde{\boldsymbol{v}}\right)_{\Omega_0^2}\\
&+\left(\nabla_{\boldsymbol{x}}^{\mathrm{sym}}\widetilde{\boldsymbol{w}},\frac{1}{\rho_1}\boldsymbol{F}\boldsymbol{S}\boldsymbol{F}^T\right)_{\Omega_0^2}\\
&+\left(\widetilde{\boldsymbol{w}},\left(1-\frac{\rho_2}{\rho_1}\right)\boldsymbol{g}\right)_{\Omega_0^2}\ .
\end{aligned}
\tag{10}
$$

Note that $\widetilde{p}$ and p' have been redefined by dividing through the fluid density ρ_1 as in [14], $\nu=\mu/\rho_1$ is the kinematic viscosity of the fluid, and $\boldsymbol{h}$ is the prescribed traction vector.

2.2. Semi-discrete formulation

In order to perform space discretization, let us start considering a set of NURBS basis functions $\{N_A^{ED}(\boldsymbol{x})\}_{A=1}^{n_{ED}}$ defined over the physical domain Ω. Thanks to the stabilized VMS technique, the same scalar discrete space spanned by the above-mentioned basis functions can be used to represent the weight and trial functions for $\widetilde{p}$ and each Cartesian component of $\widetilde{\boldsymbol{v}}$ without the need to use a stable pressure-velocity combination in the sense of Babŭska and Brezzi [29, 30]. Therefore, we define

$$
\boldsymbol{v}^h(\boldsymbol{x},t)=\sum_{A=1}^{n_{ED}}\boldsymbol{v}_A(t)N_A^{ED}(\boldsymbol{x});\quad p^h(\boldsymbol{x},t)=\sum_{A=1}^{n_{ED}}p_A(t)N_A^{ED}(\boldsymbol{x});
\tag{11}
$$

$$
\boldsymbol{w}^h(\boldsymbol{x})=\sum_{A=1}^{n_{ED}}\boldsymbol{w}_A N_A^{ED}(\boldsymbol{x});\quad q^h(\boldsymbol{x})=\sum_{A=1}^{n_{ED}}q_A N_A^{ED}(\boldsymbol{x})\ .
\tag{12}
$$

Analogously, let us consider a set of NURBS basis functions $\{N_B^{LD}(\boldsymbol{X})\}_{B=1}^{n_{LD}}$ defined over the reference domain Ω_0^2. The scalar discrete space spanned by these basis functions will be used to represent the trial functions for each Cartesian component of the displacement

$$
\boldsymbol{u}_2^h(\boldsymbol{X},t)=\sum_{B=1}^{n_{LD}}\boldsymbol{u}_B(t)N_B^{LD}(\boldsymbol{X})\ .
\tag{13}
$$

Eq. (3) will also be discretized using the space spanned by $\{N_B^{LD}(\boldsymbol{X})\}_{B=1}^{n_{LD}}$. Since we discretize this equation using collocation, we just need to define a set of collocation points. We will follow the standard practice in IGA and use Greville points (denoted by $\{\boldsymbol{\tau}_i\}_{i=1}^{n_{LD}}$) as our collocation sites. We note, however, that a new set of points referred to as Cauchy-Galerkin points has been recently proposed [31]. Cauchy-Galerkin points have the potential to render the Galerkin solution exactly with a computational cost of one evaluation of the basis functions per degree of freedom.

We will use the isoparametric concept in order to build computational meshes in Ω and Ω_0^2. These meshes will be called Eulerian mesh and Lagrangian mesh, respectively. The Lagrangian mesh can be pushed forward to physical space using the discrete mapping $\boldsymbol{\varphi}^h(\boldsymbol{X}, t)$. This mesh moves on top of the Eulerian mesh in a nonconforming fashion.

To finish the spatial discretization, we express the fine-scale velocity and fine-scale pressure in terms of their coarse-scale counterparts as

$$\boldsymbol{v}' = -\tau_M \left(\frac{\partial \boldsymbol{v}^h}{\partial t} + \boldsymbol{v}^h \cdot \nabla_{\boldsymbol{x}} \boldsymbol{v}^h + \nabla_{\boldsymbol{x}} p^h - \nu \Delta_{\boldsymbol{x}} \boldsymbol{v}^h - \boldsymbol{g} \right) , \tag{14}$$

$$p' = -\tau_C \nabla_{\boldsymbol{x}} \cdot \boldsymbol{v}^h , \tag{15}$$

where τ_M and τ_C are the stabilization parameters defined as in [32], viz.,

$$\tau_M = \left(s^h(\boldsymbol{x}, t) \left(\frac{4}{\Delta t^2} + \boldsymbol{v}^h \cdot \boldsymbol{G} \boldsymbol{v}^h + C_I \nu^2 \boldsymbol{G} : \boldsymbol{G} \right) \right)^{-\frac{1}{2}} , \tag{16}$$

$$\tau_C = \left(\tau_M \mathrm{tr} \boldsymbol{G} \right)^{-1} . \tag{17}$$

Here, $\mathrm{tr}(\cdot)$ is the trace operator, $\boldsymbol{G} = [G_{ij}] = \sum_{k=1}^{3} \frac{\partial \xi_k}{\partial x_i} \frac{\partial \xi_k}{\partial x_j}$, $\boldsymbol{x}(\boldsymbol{\xi})$ is the geometrical mapping of the Eulerian mesh, and C_I is a positive constant. Following [33], we take $C_I = 36$ in our simulations.

The scalar function $s^h(\boldsymbol{x}, t)$ that appears in Eq. (16) is a dimensionless scaling factor introduced in the context of immersed FSI methods in [32]. To the best of our knowledge, all available immersed FSI methods assume the pressure field to be continuous across the fluid-solid interface. However, the exact solution to the problem may have discontinuous pressure at the interface. Therefore, the discrete pressure spaces of immersed FSI methods have poor approximation properties at the interface leading to lack of local mass conservation in this region for practical levels of refinement. This issue is particularly notorious for thin solids where solutions with large pressure jumps at the interface are common. As in [32, 34, 35], the above-mentioned scaling factor is introduced to locally weaken the influence of the poorly approximated pressure gradient in Eq. (14) and locally enhance the penalization of volume change due to the inverse relation between τ_M and τ_C. The function $s^h(\boldsymbol{x}, t)$ belongs to the discrete space spanned by $\{N_A^{ED}(\boldsymbol{x})\}_{A=1}^{n_{ED}}$, i.e., $s^h(\boldsymbol{x}, t) = \sum_{A=1}^{n_{ED}} s_A(t) N_A^{ED}(\boldsymbol{x})$. We set the control variables associated with basis functions that intersect the interface to the same value $s_I >> 1$ and the remaining control variables are set to 1.

Finally, the semi-discrete formulation of our immersed method can be stated as: Find $\boldsymbol{v}^h$, p^h, and $\boldsymbol{u}_2^h$ such that, for all $\boldsymbol{w}^h$ and q^h defined as in Eq. (12) and all $\{\boldsymbol{\tau}_i\}_{i=1}^{n_{LD}}$,

$$B^{ED}\left(\{\boldsymbol{w}^h, q^h\}, \{\boldsymbol{v}^h + \boldsymbol{v}', p^h + p'\} \right) - L^{ED}\left(\boldsymbol{w}^h \right) + B^{LD}\left(\boldsymbol{w}^h, \boldsymbol{v}^h; \boldsymbol{u}_2^h \right) = 0 , \tag{18}$$

$$\frac{\partial \boldsymbol{u}_2^h}{\partial t}(\boldsymbol{\tau}_i, t) = \boldsymbol{v}^h(\boldsymbol{\varphi}^h(\boldsymbol{\tau}_i, t), t) . \tag{19}$$

Remark 1:

The integrals of Eq. (10) demand special attention. Gaussian quadrature rules defined on the elements of the Lagrangian mesh are used to compute those integrals. However, there are functions in the integrands, namely, the weighting functions $\boldsymbol{w}^h$, the coarse-scale velocity $\boldsymbol{v}^h$, and their first derivatives, which are defined on the Eulerian mesh instead of the Lagrangian mesh. Therefore, for a certain Gauss point with parametric coordinates $\boldsymbol{\xi}_G^{LD}$ in the Lagrangian mesh, we first need to compute its physical location $\boldsymbol{x}_G$ using the geometrical mapping of the Lagrangian mesh, i.e, $\boldsymbol{\varphi}^h$. Then, we have to invert the geometrical mapping of the Eulerian mesh in order to obtain the parametric coordinates $\boldsymbol{\xi}_G^{ED}$ associated with the physical point $\boldsymbol{x}_G$ in the Eulerian mesh. Once we know $\boldsymbol{\xi}_G^{ED}$, we can evaluate functions that are defined on the Eulerian mesh using standard procedures of finite elements. Note that an analogous situation occurs when we try to evaluate Eq. (19) at a certain collocation point.

Remark 2:

The above-explained integration process used for B^{LD} is suboptimal because the functions that are defined on the Eulerian mesh may have lines of reduced continuity in the interior of the integration regions which are the elements of the Lagrangian mesh. However, the higher inter-element continuity of spline functions is used in order to alleviate this issue.

Remark 3:

As reported in the literature [36, 37], when piecewise linear Lagrange polynomials are used in immersed FSI, the element size of the Lagrangian mesh is required to be at least twice as small as the element size of the Eulerian mesh in order to achieve stable solutions. Our numerical findings reveal that the use of higher order splines permits to obtain stable solutions utilizing the same element size in both meshes.

2.3. Time discretization and FSI solution strategy

We discretize the first order time derivatives that appear in our immersed FSI formulation using the Generalized-α technique [38], which is an implicit and second-order accurate method with control over the dissipation of high-frequency modes. Once the time discretization has been sorted, we are left with a system of nonlinear algebraic equations that relates the unknown control variables of the Eulerian velocity, Eulerian pressure, and Lagrangian displacement at time t^{n+1} to the known control variables at time t^n. In order to compute the unknown control variables at time t^{n+1}, we derive two separate tangent matrices. One tangent matrix for the linear momentum and mass conservation residuals ($\boldsymbol{R}_{MC}$) where the Lagrangian control variables ($\boldsymbol{U}_{LD}$) are considered to be constant and another tangent matrix for the kinematic equation residual ($\boldsymbol{R}_K$) where the Eulerian control variables ($\boldsymbol{U}_{ED}$) are considered to be constant. The tangent matrix associated to $\boldsymbol{R}_K$ is computed by linearizing the residual with respect to the physical values of the Lagrangian displacements at collocation points. The reason is that this tangent matrix is much sparser than that obtained by linearizing with respect to the control variables. Actually, by using

the advocated tangent matrix we only need to solve an independent $d \times d$ linear system for each collocation point. Furthermore, we apply the so-called block-iterative approach for solving these two subproblems [39, 40]. This fully-implicit approach boils down to finding a root of the residuals $\boldsymbol{R}_{MC}$ and $\boldsymbol{R}_K$ given initial guesses for $\boldsymbol{U}_{ED}$ and $\boldsymbol{U}_{LD}$ following the next steps:

1. Compute the residual $\boldsymbol{R}_K$ by evaluating Eq. (19) at each collocation point and the corresponding tangent matrix by linearizing with respect to the physical values of the displacements at collocation points.
2. Solve a $d \times d$ linear system for each collocation point where the unknowns are the physical values of the displacement at the collocation point.
3. Compute the Lagrangian control variables from the displacement physical values at the collocation points. This is done collocating Eq. (13) and solving a linear system. Note that the matrix of this linear system depends only on the parametric configuration of the Lagrangian mesh. Therefore, we just need to compute the matrix once, store it, and use it in each Newton-Raphson iteration of each time step. This finalizes the update of the Lagrangian control variables $\boldsymbol{U}_{LD}$.
4. Assemble the residual $\boldsymbol{R}_{MC}$. We recall that the assembly of $\boldsymbol{R}_{MC}$ is not standard since it requires to go from the parametric coordinates of a quadrature point in the Lagrangian mesh to its associated parametric point in the Eulerian mesh at the proper time level. This process was explained in detail in [24] and avoids errors associated with either interpolating or projecting functions from the Eulerian mesh to the Lagrangian mesh.
5. Assemble an approximate tangent matrix $\boldsymbol{K}_{MC} \simeq \dfrac{\partial \boldsymbol{R}_{MC}}{\partial \boldsymbol{U}_{ED}}$. We are using the approximation to the tangent matrix given in [14].
6. Solve the linear system $\boldsymbol{K}_{MC} \Delta \boldsymbol{U}_{ED} = -\boldsymbol{R}_{MC}$.
7. Update the Eulerian control variables $\boldsymbol{U}_{ED} \leftarrow \boldsymbol{U}_{ED} + \Delta \boldsymbol{U}_{ED}$.

This process is repeated until each component of $\boldsymbol{R}_{MC}$ and $\boldsymbol{R}_K$ is sufficiently converged. In particular, in all the simulations performed in this paper, we stop the iterative process when the L^2-norm of each residual component has decreased more than four orders of magnitude. To linearize the residual $\boldsymbol{R}_{MC}$, we use a Newton-Raphson algorithm with line search [41]. Regarding the linear solver for Step 6, we use the generalized minimum residual method (GMRES) [42, 43] with relative tolerance set to 10^{-5}. The use of the function s^h in Eq. (16) increases the condition number of $\boldsymbol{K}_{MC}$ for large values of the parameter s_I. This makes it quite challenging to find a suitable preconditioner. Our implementation is built on top of PetIGA [44, 45, 46], which gives us direct access to the preconditioners of the scientific librery PETSc [47, 48]. For parallel MPI simulations with a relatively low number of processors (up to 32 or 64 cores depending on the size of the problem), the best performance is given by the restricted additive Schwarz preconditioner with the following options: LU decomposition as subdomain solver, one subdomain per processor, and nested dissection as matrix reordering technique so as to reduce the ratio of fill-ins

and enhance the stability of the LU decomposition. However, for parallel MPI simulations with a larger number of processors (we have tried up to 256 cores), the best performance is given by the block Jacobi preconditioner with the following options: LU decomposition as subdomain solver, two subdomains per processor so as to enhance the communication between subdomains [49], and nested dissection as matrix reordering method.

Remark 4:

In various of the simulations presented in the following Sections, we will have several solids embedded in the domain Ω. The aforementioned block-iterative technique is highly efficient since we just have to repeat Steps 1, 2, and 3 for each solid and assemble its contribution to $\boldsymbol{R}_{MC}$ in Step 4.

3. Numerical simulations of capsules and compound capsules in shear flow

The velocity field of a linear shear flow in the x direction has the form $\boldsymbol{v}_{shear} = (\dot{\gamma}z, 0, 0)$ where $\dot{\gamma}$ is the constant shear rate that characterizes the flow's strength. For low Reynolds numbers, shear flows can be produced in rectangular and parallelogram geometries by setting velocity boundary conditions compatible with $\boldsymbol{v}_{shear}$ and letting the Navier-Stokes equations evolve in time. In this Section, we follow a similar approach, but we embed capsules and compound capsules in the flow. The goal is to study their motions and how they modify the velocity field.

3.1. Qualitative description of the motion of a capsule

A capsule, which is used as our RBC mimic, is composed by a very thin membrane with a fluid inside. From theoretical and experimental developments, it is known that the following dimensionless numbers control the dynamics of an individual capsule to a significant extent:

- The swelling degree of the membrane (Δ_m). In a three-dimensional setting, the swelling degree is defined as the ratio of the volume enclosed by the membrane (V_e) to the volume of a sphere with the membrane external area (A_c), that is, $\Delta_m = 6\sqrt{\pi}V_e/\sqrt{A_c^3}$. Its two-dimensional counterpart is defined as the ratio of the area enclosed by the membrane (A_e) to the area of a circle with the membrane perimeter (P_c), that is, $\Delta_m = 4\pi A_e/P_c^2$.

- The confinement degree (χ). This quantity is the ratio of the effective membrane radius (R_0) to the channel half-width. In a three-dimensional setting, R_0 is defined as the radius of a sphere with the membrane external area, i.e., $R_0 = \sqrt{A_c/4\pi}$. Analogously, in a two-dimensional setting, R_0 is the radius of a circle with the membrane perimeter, i.e., $R_0 = P_c/2\pi$[†].

[†]Some authors define an effective radius over the volume/area enclosed by the membrane in a three-/two-dimensional setting, respectively.

- The viscosity contrast (Λ). The membrane encloses an internal fluid of viscosity μ_i and it is suspended in an outer fluid of viscosity μ_o, thus defining the viscosity contrast as $\Lambda = \mu_i/\mu_o$. In this paper, we consider internal and outer fluids with the same viscosity μ, therefore, $\Lambda = 1$.

- The surface capillary number of the membrane (C_a^m). The parameter C_a^m quantifies the relative strength of the viscous forces exerted by the external fluid to the elastic forces exerted by the membrane. Its precise form is given by $C_a^m = \mu_o \dot{\gamma} R_0/hG_m$, where $\dot{\gamma}$ is the average shear rate, h is the membrane thickness, and G_m is the membrane shear modulus.

- The Reynolds number (R_e). The parameter R_e quantifies the relative strength of the inertial forces to the viscous forces exerted by the external fluid. The relevant Reynolds number for this problem is $R_e = \rho_1 \dot{\gamma} R_0^2/\mu_o$.

A single capsule in shear flow exhibits two main types of motion, viz., tank-treading and tumbling. These two motions were first observed in experiments with RBCs in shear flow in [50, 51, 52]. In the tank-treading (TT) motion, the RBC membrane adopts a nearly constant orientation with the flow direction and rotates around the hemoglobin like the tread of a tank. The two main physical quantities that characterize a TT motion are the inclination angle of the membrane with the flow direction (θ) and the frequency of the TT motion (ω_{TT}). In the tumbling (TU) motion, the RBC rotates as a whole around its center of gravity. The main physical quantity that characterizes a TU motion is its frequency (ω_{TU}). For capsules, the transition from TT to TU is mainly determined by the viscosity contrast. For $\Lambda = 1$, TT motion is expected.

3.2. Qualitative description of the motion of a compound capsule

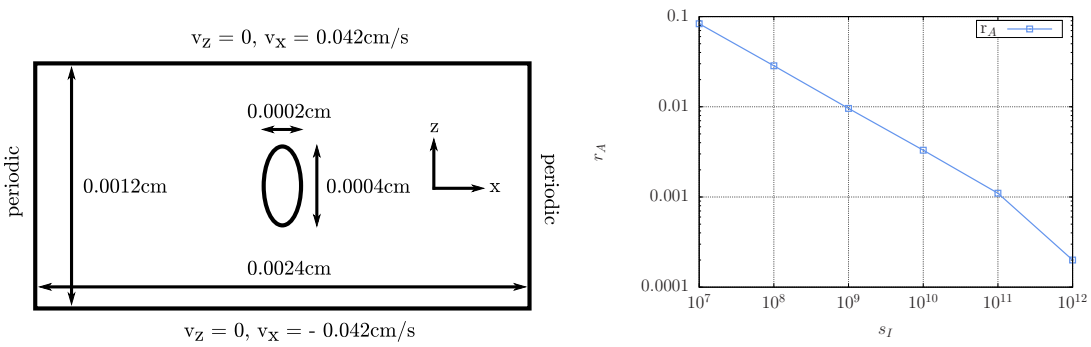
A compound capsule, which is proposed here as a CTC mimic, is composed by a very thin membrane, an inclusion (i.e., a bulky solid) and a fluid between the membrane and the inclusion. The dynamics of a compound capsule are controlled by the quantities described in Section 3.1. and the following dimensionless numbers:

- The filling fraction (ϕ). In a three-dimensional setting, ϕ is the ratio of the inclusion volume to the volume enclosed by the membrane. Its two-dimensional counterpart is defined as the ratio of the inclusion area to the area enclosed by the membrane.

- The swelling degree of the inclusion (Δ_i). The definition of Δ_i is analogous to that of Δ_m.

- The bulk capillary number of the inclusion (C_a^i). The quantity C_a^i measures the relative strength of the viscous forces exerted by the internal fluid to the elastic forces exerted by the inclusion. It is defined as $C_a^i = \mu_i \dot{\gamma}/G_i$, where G_i is the inclusion shear modulus.

The numerical simulations that follow are aimed at gaining a deeper understanding of the dynamics of a compound capsule in shear flow.



(a) Problem setup (b) Relative area change

Figure 1: Parameter sweep of s_I to asses mass conservation errors near the fluid-solid interface. a) Geometrical description of the considered problem together with the boundary conditions applied. b) Relative area change of an elliptic capsule in shear flow at $t = 1.0$ s with $\Delta t = 0.0005$ s.

3.3. Adjusting s_I so as to enforce the incompressibility constraint at the interface

As explained in Section 2.2 and following [32], we modify the VMS stabilization close to the fluid-solid interface introducing the dimensionless parameter s_I in order to compensate the poor approximation properties of the pressure field in that region. In order to decide a suitable value for s_I, we measure the area change of a two-dimensional capsule for different values of s_I.

The domain Ω is a rectangle of 0.0024×0.0012 cm^2. Periodic boundary conditions are applied in the horizontal direction. We use Dirichlet boundary conditions in the top and bottom walls as indicated in Fig. 1 a). The described boundary conditions lead to a shear rate $\dot{\gamma} = 70$ s^{-1}, which is within the range found in microcirculation [53]. An elliptic capsule is initially placed at the center of the domain with its longer axis perpendicular to the flow direction [see Fig. 1 a)]. The semiaxes of the ellipse are $a = 0.0002$ cm and $b = 0.0001$ cm, which lead to $R_0 = 0.0001542$ cm. The thickness of the membrane is $h = 5 \cdot 10^{-7}$ cm. The remaining physical parameters are taken as follows: $\mu = 0.012$ g/(cm s), $\rho_1 = 1.025$ g/cm^3, $\rho_2 = 1.1$ g/cm^3, $G_m = 7500$ dyn/cm^2, and $\boldsymbol{g} = (0, 0)$. With these parameters, the above-mentioned dimensionless numbers are: $\Delta_m = 0.841$, $\chi = 0.257$, $\Lambda = 1$, $C_a^m = 0.0345$, and $R_e = 0.000142$.

Here and in the remaining simulations of this paper, our solid material model is given by the second Piola-Kirchhoff stress tensor

$$\boldsymbol{S} = GJ^{-2/d}\left(\boldsymbol{I} - \frac{1}{d}\mathrm{tr}(\boldsymbol{C})\boldsymbol{C}^{-1}\right) , \tag{20}$$

where G is the shear modulus and $\boldsymbol{C} = \boldsymbol{F}\boldsymbol{F}^T$ is the Cauchy-Green deformation tensor.

An Eulerian mesh with 256×128 cubic NURBS elements is used to represent the rectangular domain Ω. To represent the capsule we use NURBS basis functions of second order or higher, which, unlike Lagrange polynomials, can represent a hollow ellipse exactly.

In this simulation, we use a Lagrangian mesh with 1×208 quadratic NURBS elements. The time step is $\Delta t = 0.0005$ s.

We define the relative area change of the capsule at time t as

$$r_A(t) = \frac{\int_{\Omega_t^2} \mathrm{d}\Omega_t^2 - \int_{\Omega_0^2} \mathrm{d}\Omega_0^2}{\int_{\Omega_0^2} \mathrm{d}\Omega_0^2}. \tag{21}$$

In this simulation, the capsule acquires a nearly constant inclination angle at time $t = 0.15$ s. At time $t = 1$ s, the membrane particles have already completed more than three turns in their TT motion. Fig. 1 b) shows the value of r_A for different values of s_I at time $t = 1$ s. As expected, the spurious area change decreases as we rise the dimensionless parameter s_I. As we increase the value of s_I, the number of linear iterations needed per nonlinear iteration increases because the condition number of the tangent matrix becomes greater. However, the number of nonlinear iterations per time step decreases because larger values of s_I render a less stiff residual of the mass conservation equation, which is usually the bottleneck for nonlinear convergence. In particular, for this case, which is solved using the restricted additive Schwarz preconditioner, the computational time was lowest for $s_I = 10^{10}$. Based on Fig. 1 b), we believe that $s_I = 10^{10}$ produces mass conservation errors which are sufficiently small for our target applications. Therefore, we will use this value of s_I in all the simulations of shear flow unless otherwise specified. Consistently with [32], we find that large errors in the incompressibility constraint may even alter the qualitative character of the numerical solutions. For example, for $s_I \leq 10^5$, the capsule no longer undergoes a classical TT motion. To test and validate an immersed FSI algorithm, we recommend to check the errors incurred in the incompressibility constraint near the fluid-solid interface.

Under physiological conditions, both the RBC volume and external area are known to remain nearly constant. We are not enforcing the capsule external area to be constant in our formulation. However, we will measure its time evolution in our simulations. In order to do so, we define the relative perimeter change of the capsule in our two-dimensional setting as

$$r_P(t) = \frac{\int_{\Gamma_t^2} \mathrm{d}\Gamma_t^2 - \int_{\Gamma_0^2} \mathrm{d}\Gamma_0^2}{\int_{\Gamma_0^2} \mathrm{d}\Gamma_0^2}. \tag{22}$$

In the simulations shown in Section 3, the relative perimeter change is always lower than 0.04. We believe that this value is acceptable for the purposes of this paper. There are capsule formulations based on shells that impose the constraint of external area in their formulations [54, 55]. However, as pointed out in [56], when these models are used in immersed FSI, this constraint is usually loosened in order to achieve convergence of the FSI scheme.

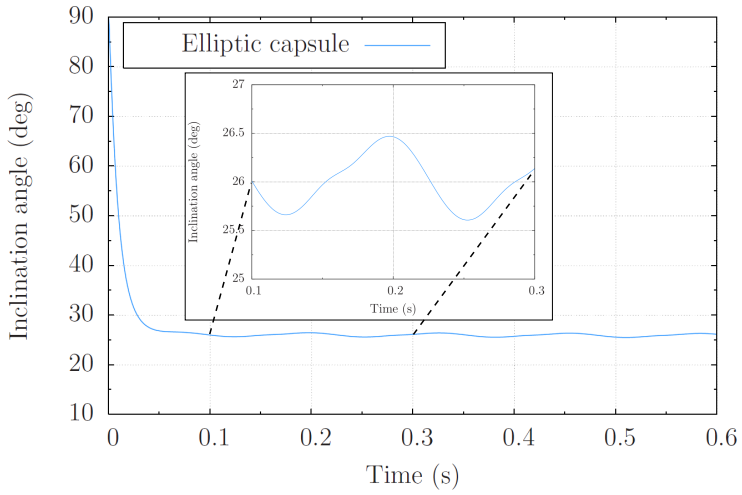


Figure 2: Time evolution of the inclination angle. In this case, there is an oscillation with an amplitude smaller than 1° around the value 26.06°

3.4. Spatial and temporal resolution study

In this Section, we perform a mesh independence study for the problem stated in Section 3.3. The quantities of interest will be the inclination angle and the TT frequency for different meshes and time steps. In our simulations, the inclination angle is computed as the angle between the minor principal axis of inertia of the capsule and the flow direction. The TT frequency is computed as the average frequency with which the particles of the membrane move.

When the inclination angle of a RBC is experimentally measured, it oscillates between 0.1° and 2° around a certain average value. This oscillation was initially attributed to thermal fluctuations [5]. However, this oscillation was also found in numerical simulations with capsules that do not take into account thermal fluctuations [57, 56, 58] and it is also present in the TT motions shown in this Section; see Fig. 2. To compute the inclination angle, we monitor its time evolution and when it enters an oscillatory regime, we take the mean value.

We consider quadratic and cubic Eulerian meshes and perform h-refinement. In all cases, we define a quadratic Lagrangian mesh with slightly smaller element size than that of the Eulerian mesh under consideration[‡]. The results are shown in Table 1. First of all, the data shows that, for the time steps used, the time integration is not compromising the accuracy of the solution. Second of all, for a given number of elements, cubics are more accurate than quadratics. This was expected due to their higher power of approximation [59, 33]. However, the significant differences observed in the data may be also attributed to the suboptimal integration issue mentioned in Section 2.2, which is alleviated using cubics.

[‡]We have also tried cubic Lagrangian meshes and Lagrangian meshes with element size twice as small as that of the Eulerian mesh (data not shown). Neither θ nor ω_{TT} changed with respect to the results shown in Table 1.

# elements (Eulerian mesh)	Degree (Eulerian mesh)	Δt (s)	θ (°)	ω_{TT} (s^{-1})
96×48	2	0.001	17.81	3.20
96×48	3	0.001	23.95	3.54
128×64	2	0.001	22.14	3.46
128×64	2	0.0005	22.24	3.45
128×64	3	0.001	25.22	3.58
128×64	3	0.0005	25.22	3.58
192×96	2	0.0005	24.45	3.59
192×96	3	0.0005	25.77	3.78
256×128	2	0.0005	25.24	3.72
256×128	2	0.00025	25.24	3.72
256×128	3	0.0005	25.99	3.84
256×128	3	0.00025	25.99	3.84
384×192	2	0.0005	25.80	3.84
384×192	2	0.0001	25.80	3.84
384×192	3	0.0005	26.06	3.88
384×192	3	0.0001	26.06	3.88
512×256	2	0.00025	25.87	3.87
512×256	3	0.00025	26.09	3.90

Table 1: Spatial and temporal resolution study. We analyze the influence of the discretization on the inclination angle and TT frequency.

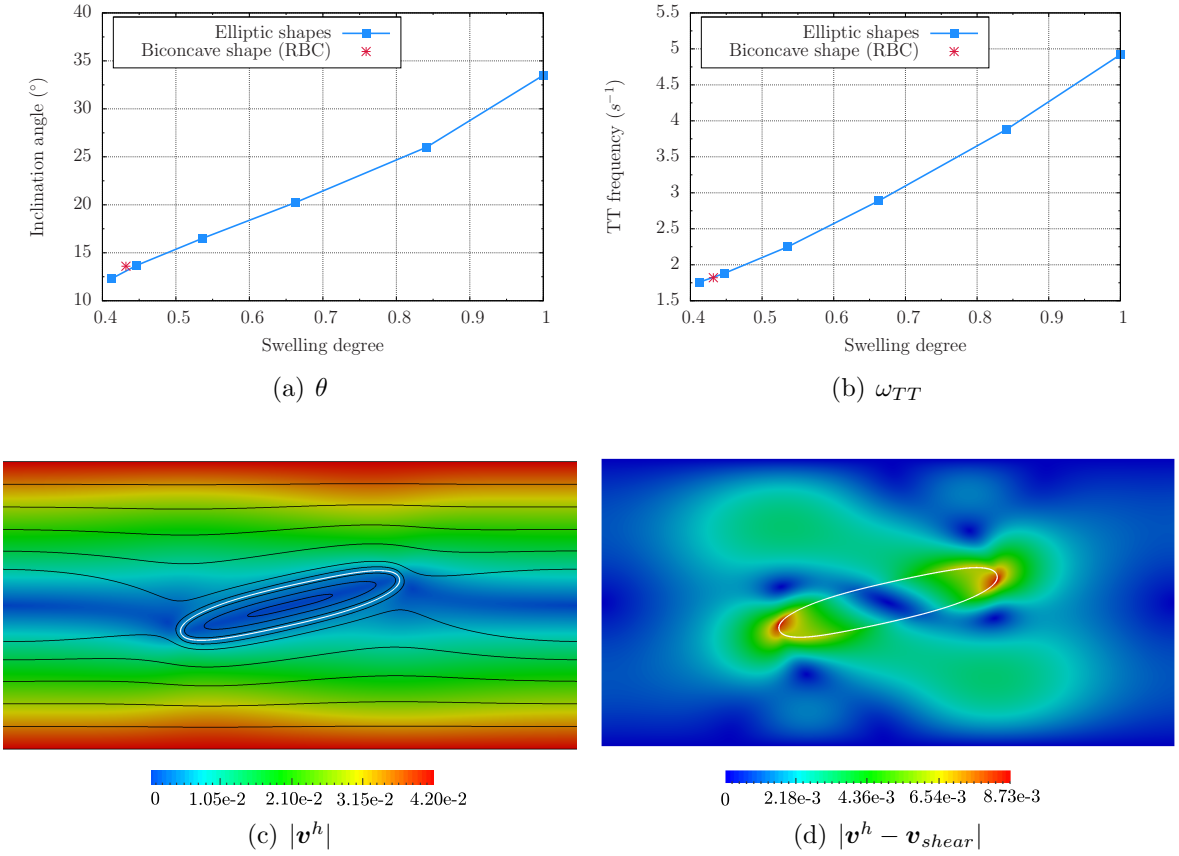
Finally, the small differences in θ and ω_{TT} for finer levels of refinement suggest that a converged result has been obtained.

Based on this mesh independence study, we will use an Eulerian mesh with 384×192 cubic elements, a quadratic Lagrangian mesh with a slightly smaller element size than that of the Eulerian mesh, and $\Delta t = 0.0005$ s in all the two-dimensional simulations of Section 3.5 and Section 3.6.

3.5. Dynamics of a capsule

We consider capsules with different geometries and embed them in the shear flow defined in Section 3.3. We start working with six elliptic capsules with different swelling degrees. In order to create these capsules, we vary the longer semiaxis of the capsule defined in Section 3.3 and keep unchanged all the other geometrical and mechanical parameters of the capsule. We consider the following values for a: 0.0001 cm, 0.0002 cm, 0.0003 cm, 0.0004 cm, 0.0005 cm, and 0.00055 cm. The computed inclination angle and TT frequency for each capsule are plotted in Fig. 3 a) and b), respectively. Consistently with the results reported in [60], both θ and ω_{TT} increase with the value of the swelling degree.

Now, we repeat the analysis considering a capsule with the biconcave shape that RBCs have in a quiescent fluid, which is defined in [55]. The computed θ and ω_{TT} for this new initial shape are plotted in Figs. 3 a) and b) with a red star. The results suggest that



(a) θ (b) ω_{TT}

(c) $|\boldsymbol{v}^h|$ (d) $|\boldsymbol{v}^h - \boldsymbol{v}_{shear}|$

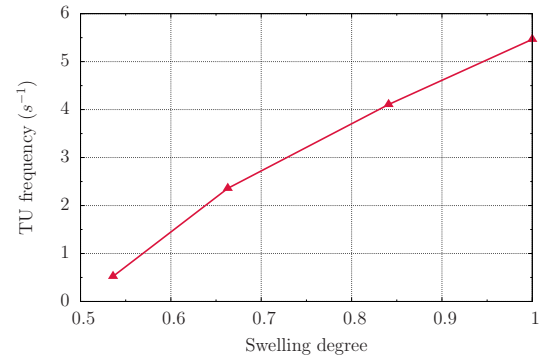
Figure 3: Capsules with different shapes under shear flow. (a) Inclination angle of a capsule as a function of the swelling degree. (b) Tank-treading frequency of a capsule as a function of the swelling degree. (c) Velocity magnitude along with the flow streamlines at time $t = 0.5$ s for the capsule with initial biconcave shape. The black lines represent the streamlines and the white line denotes the capsule. There is a vortex whose center is located at the center of gravity of the capsule. (d) Perturbations introduced by the biconcave capsule with respect to the shear component at time $t = 0.5$ s.

the swelling degree characterizes quite accurately the behavior of capsules in the nonlinear regime independently of their initial shape.

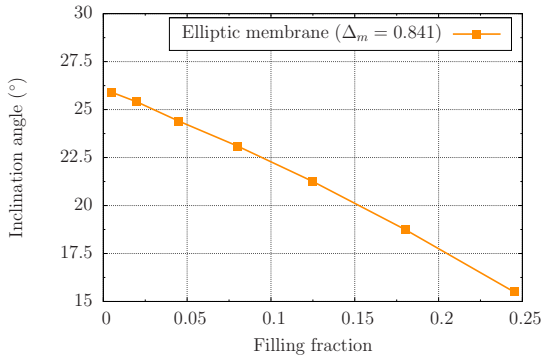
Fig. 3 c) displays a contour plot of the velocity magnitude ($|\boldsymbol{v}^h|$) with flow streamlines for the biconcave shape at time $t = 0.5$ s. The streamlines close to the capsule are parallel to the membrane which is a consequence of its tank-treading motion. Fig. 3 d) plots the velocity field resulting from our FSI simulation minus the shear component ($|\boldsymbol{v}^h - \boldsymbol{v}_{shear}|$) in order to show how the presence of the capsule modifies the flow conditions.

3.6. Dynamics of a compound capsule

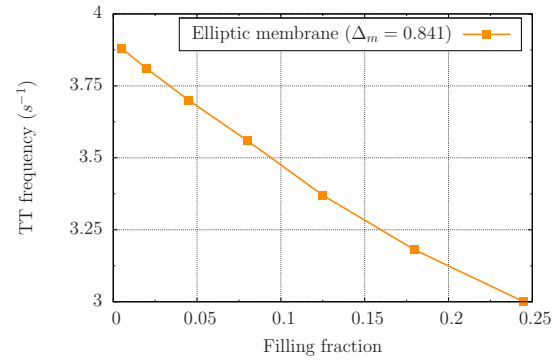
Cells with nucleus, e.g., WBCs, have been often modeled as bulky solids [61] or fluid-filled membranes [62, 63]. However, these models fail to accurately reproduce the basic dynamics of nucleated cells [64]. For example, we have embedded bulky ellipses with



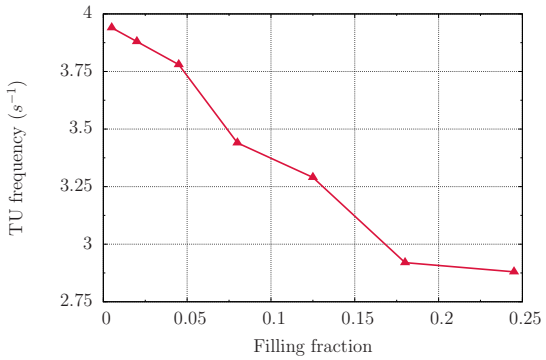
(a) ω_{TU} of bulky solids



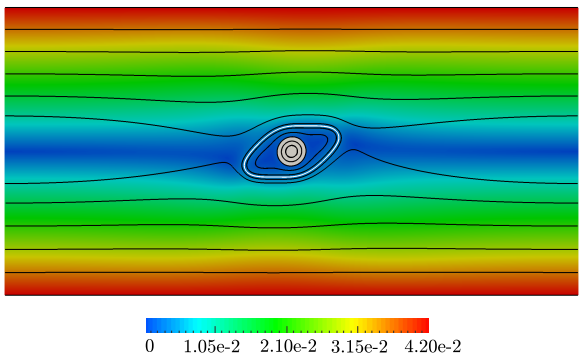
(b) θ of the membrane (compound capsule)



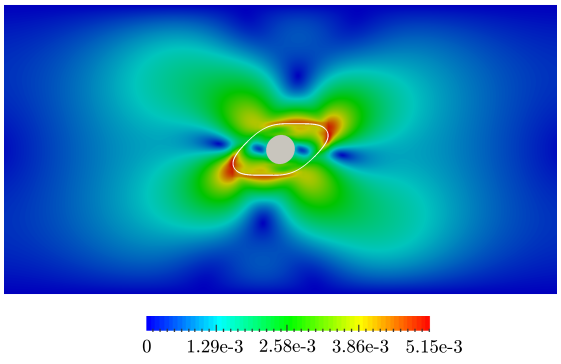
(c) ω_{TT} of the membrane (compound capsule)



(d) ω_{TU} of the inclusion (compound capsule)

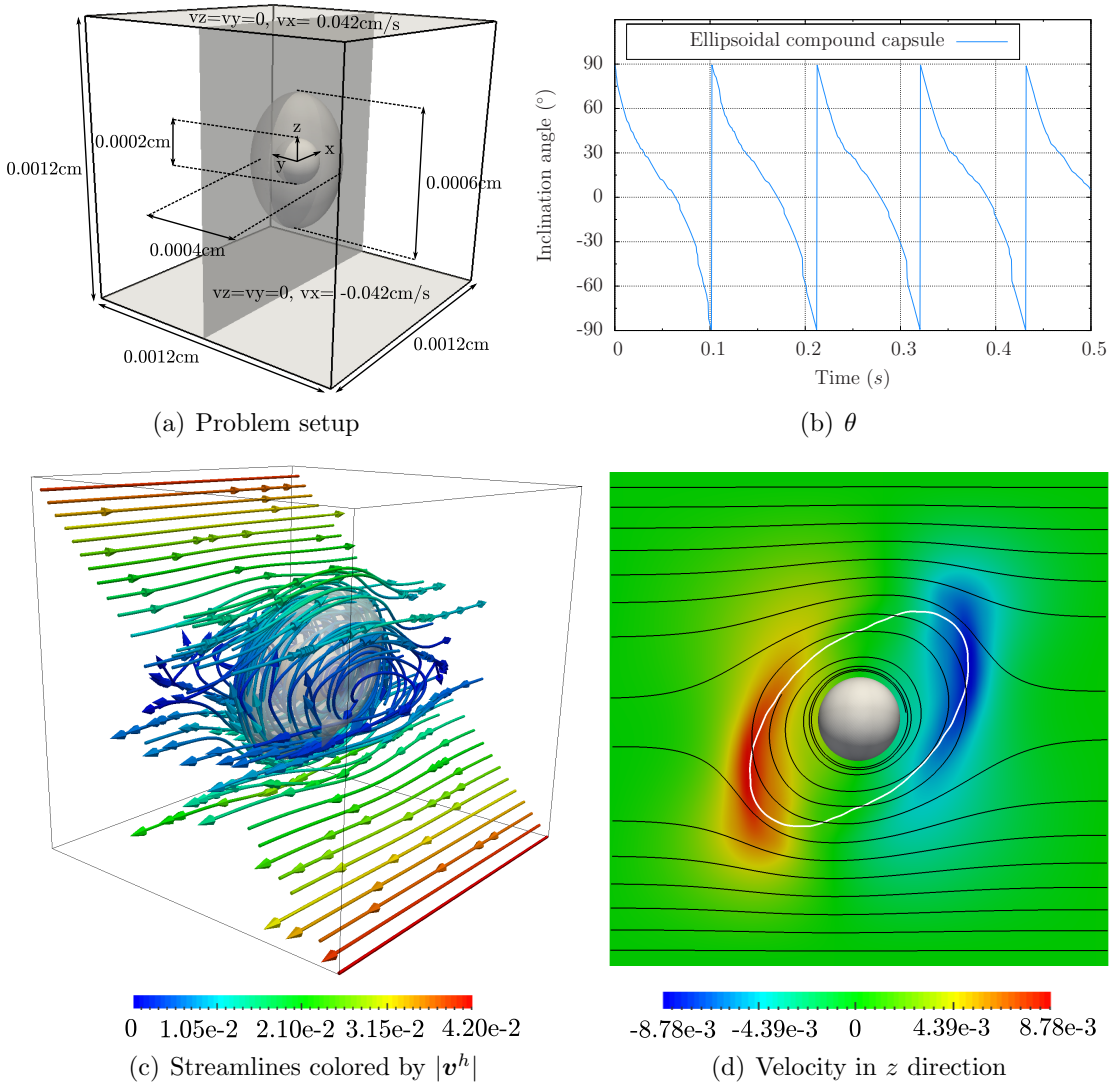


(e) $|\boldsymbol{v}^h|$



(f) $|\boldsymbol{v}^h - \boldsymbol{v}_{shear}|$

Figure 4: Bulky solids and compound capsules under shear flow. (a) Tumbling frequency of a bulky solid under shear flow as a function of the swelling degree. (b) Inclination angle of the membrane of a compound capsule as a function of the filling fraction. (c) Tank-treading frequency of a compound capsule membrane as a function of the filling fraction. (d) Tumbling frequency of an inclusion under pure shear flow as a function of the filling fraction. (e) Velocity magnitude along with the flow streamlines at time $t = 0.5$ s for the compound capsule with $\phi = 0.180$. The streamlines near the boundary of the compound capsule are parallel to the membrane due to its tank-treading motion. (f) Perturbations introduced by the compound capsule with respect to the shear component at time $t = 0.5$ s for the compound capsule with $\phi = 0.180$.



(a) Problem setup (b) θ

(c) Streamlines colored by $|\boldsymbol{v}^h|$ (d) Velocity in z direction

Figure 5: A compound capsule undergoing a TU motion in a three-dimensional setting. (a) Geometrical description of the problem along with Dirichlet boundary conditions applied on the two shaded faces of the cube. Periodic boundary conditions are applied in the other four faces. The shaded plane, i.e., the plane $y = 0$ is used to plot the z component of the velocity in Fig. 5 d). (b) Time evolution of the inclination angle. (c) Streamlines colored by the velocity magnitude at time $t = 0.23$ s. A transparent membrane is plotted in order to see the streamlines and the inclusion inside of it. The boundary of Ω is represented by black lines. (d) Velocity in z direction on the plane indicated in Fig. 5 a) at time $t = 0.23$ s along with the streamlines.

different swelling degrees in the shear flow defined in Section 3.3. As expected, they all perform a TU motion whose frequency is plotted in Fig. 4 a). However, one may foresee more complicated types of motion for a cell with nucleus. For this reason, alternative models have recently been proposed such as bilamellar vesicles [65] and compound vesicles [16],

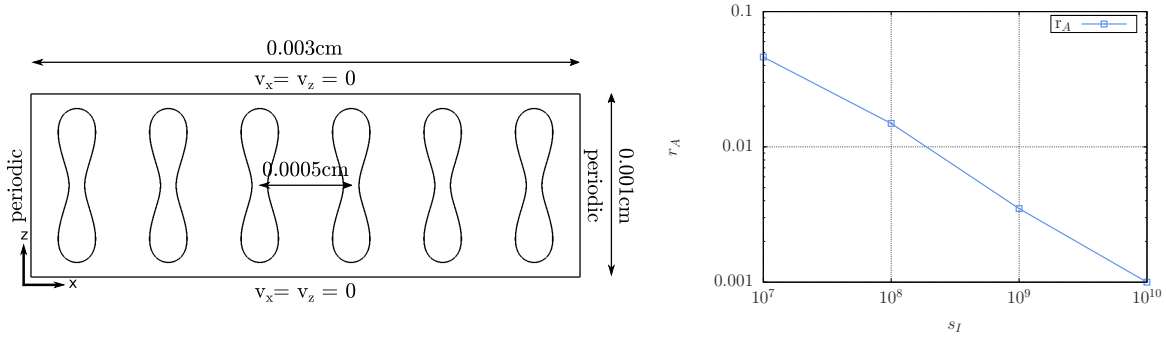
which aim at capturing the internal structure of these cells more accurately by mimicking the effect of the nucleus. In this work, we present compound capsules as the capsule counterpart of the model introduced in [16][§].

We start embedding compound capsules formed by circular inclusions ($\Delta_i = 1$) of different sizes. We consider the following values for the inclusion radius r: 0.00001 cm, 0.00002 cm, 0.00003 cm, 0.00004 cm, 0.00005 cm, 0.00006 cm, and 0.00007 cm. The shear modulus of the inclusion is $G_i = 40$ dyn/cm^2, hence $C_a^i = 0.021$. In all cases, we consider the elliptic membrane defined in Section 3.3, which leads to compound capsules with different filling fractions. In these simulations, the membrane undergoes a TT motion while the inclusion performs a TU motion. The values of θ and ω_{TT} for the membrane are plotted for each compound capsule in Fig. 4 b) and c), respectively. Consistently with [16], both the inclination angle and the TT frequency decrease with the filling fraction value. Moreover, the inclination angle and TT frequency values tend to those of a capsule as the inclusion becomes smaller, which can be seen comparing Fig. 4 with Fig. 3. The TU frequency of the inclusion is plotted in Fig. 4 d). Fig. 4 e) displays a contour plot of the velocity magnitude with the flow streamlines at time $t = 0.5$ s for the case $\phi = 0.180$. Fig. 4 f) plots $|\boldsymbol{v}^h - \boldsymbol{v}_{shear}|$ at time $t = 0.5$ s.

Finally, we consider a compound capsule in a three-dimensional setting; see 5 a). The domain Ω is a cube of side 0.0012cm. Periodic boundary conditions are applied in the two horizontal directions and Dirichlet boundary conditions are applied in the top and bottom walls so as to impose a pure shear flow with shear rate $\dot{\gamma} = 70$ s^{-1}. We embedded a compound capsule at the center of the cube. The membrane is an ellipsoid with semiaxes $a = 0.0003$ cm, $b = 0.0002$ cm, and $c = 0.0002$ cm which is initially oriented as in Fig. 5 a). The thickness of the membrane is $h = 5 \cdot 10^{-7}$ cm and its effective radius is $R_0 = 0.000232$ cm. The inclusion is a sphere with a radius $r = 0.0001$ cm. The remaining physical parameters are taken as follows: $\mu = 0.012$ g/(cms), $\rho_1 = 1.025$ g/cm^3, $\rho_2 = 1.1$ g/cm^3, $G_m = 7500$ dyn/cm^2, $G_i = 40$ dyn/cm^2. With these parameters, the dimensionless numbers are: $\Delta_m = 0.960$, $\Delta_i = 1.0$, $\chi = 0.387$, $\Lambda = 1$, $C_a^m = 0.0520$, $Re = 0.000322$, $C_a^i = 0.021$, and $\phi = 0.083$.

An Eulerian mesh with 262144 quadratic NURBS elements is used to represent the cube. To represent the solid, we use NURBS functions of second order, which are able to represent exactly both a hollow ellipsoid and a bulky sphere. We use meshes with 3072 (ellipsoid) and 7168 (sphere) quadratic NURBS elements. This problem is solved using the block Jacobi preconditioner and s_I was set to 10^9. In this computation, both the membrane and the inclusion perform a TU motion, i.e., the two of them rotate as a whole around their center of gravity. A TU motion is characterized by its average frequency which in this case is 4.68 s^{-1} for both the inclusion and the membrane. In Fig. 5 b), we show how the inclination angle varies in time. Fig. 5 c) displays the streamlines colored by the velocity magnitude in a diagonal plane of the cube and close to the compound capsule at

[§]The term compound capsule was recently used in [66]. However, following the terminology of [16, 65], the authors do not use compound capsules, but bilamellar capsules, i.e., a capsule inside another capsule.



(a) Problem setup (b) Relative area change

Figure 6: Parameter sweep of s_I. This study verifies that the mass conservation equation is adequately enforced near the fluid-solid interface. a) Geometrical description of the considered problem together with the boundary conditions imposed. b) Relative area change of RBCs in parabolic flow at $t = 1.1$ s with $\Delta t = 0.00025$ s.

time $t = 0.23$ s. Fig. 5 d) displays the z component of the velocity on the plane indicated in Fig. 5 a). This component is responsible for the TU motion of the membrane.

4. Numerical simulations of capsules and compound capsules in parabolic flow

In this Section, we study the behavior of our cell mimics in flows driven by a pressure gradient. For low Reynolds numbers and in the absence of solids this produces parabolic flows. We study how the flow is modified by capsules and compound capsules and the motions undergone by our cell mimics.

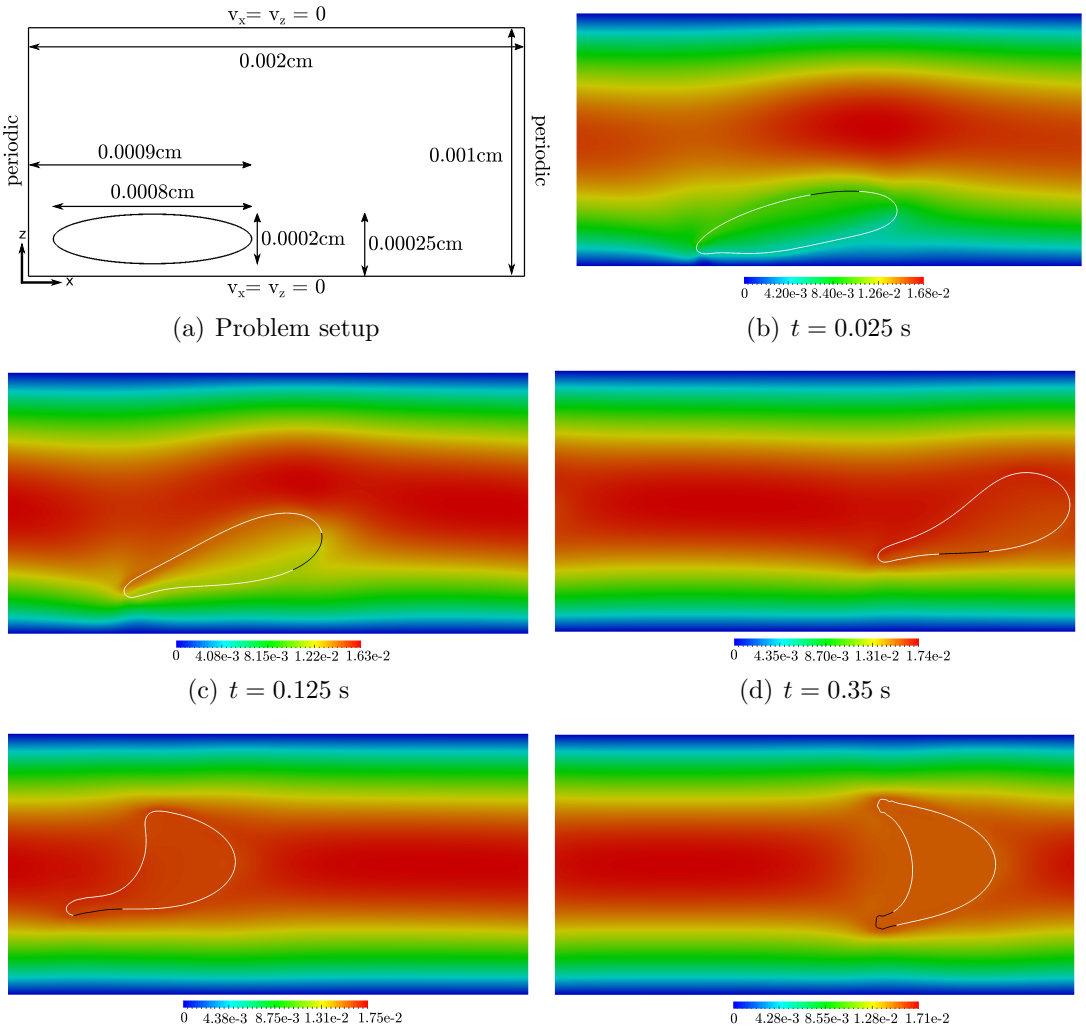
4.1. Adjusting s_I so as to enforce the incompressibility constraint at the interface

We start deciding a suitable value of s_I. Finding an adequate value of s_I is, in principle, a problem-dependent task and it should be done for each type of problem.

The domain Ω is the rectangle of 0.003×0.001 cm^2 shown in Fig. 6 a). Periodic boundary conditions are applied in the horizontal direction and no-slip boundary conditions are applied in the top and bottom walls. The pressure gradient is applied as a body force defined by $\boldsymbol{g} = (1666.66 \text{ dyn/cm}^3, 0 \text{ dyn/cm}^3)$. This leads to velocities that are within the range found in microcirculation [53]. Six capsules with the RBC shape defined in [55] are evenly distributed in the horizontal direction with its long side perpendicular to the flow direction as shown in Fig. 6 a). The thickness of the RBC is $h = 5 \cdot 10^{-7}$ cm. The remaining physical parameters are taken as follows: $\mu = 0.012$ g/(cms), $\rho_1 = 1.025$ g/cm^3, $\rho_2 = 1.1$ g/cm^3, and $G_m = 7500$ dyn/cm^2.

An Eulerian mesh with 384×128 cubic NURBS elements is used to represent the rectangle. A Lagrangian mesh with 1×328 quadratic NURBS elements is used to represent each RBC. The time step used in the computation is $\Delta t = 0.00025$ s.

In this simulation, RBCs acquire a nearly constant shape at time $t = 0.45$ s. By time $t = 1.1$ s, the cells have already looped twice the periodic domain. Fig. 6 b) plots the value



(a) Problem setup (b) $t = 0.025$ s

(c) $t = 0.125$ s (d) $t = 0.35$ s

(e) $t = 0.625$ s (f) $t = 1.5$ s

Figure 7: Capsule in parabolic flow. a) Geometrical description of the problem together with the boundary conditions applied. b)-f) Snapshots describing the lateral migration of the capsule induced by the variation in the shear rate. The capsule undergoes a transitory tank-treading motion until it gets its final parachute-like symmetric shape shown in f). A small portion of the membrane is colored in black so that the TT motion can be observed.

of r_A for different values of s_I at time $t = 1.1$ s. As in Section 3.3, the spurious area change decreases as we rise the dimensionless parameter s_I. Again, the number of linear iterations per nonlinear step increases with the value of s_I, but the number of nonlinear iterations per time step decreases with the value of s_I. In this case, the computational time was lowest for $s_I = 10^8$. However, we will use the value $s_I = 10^9$ in the remaining simulations of this Section to have a more accurate result.

In the simulations shown in Section 4, the relative perimeter change was always lower than 0.03. This implies that in this aspect, the results are even better than in the previous Section.

4.2. Dynamics of a capsule

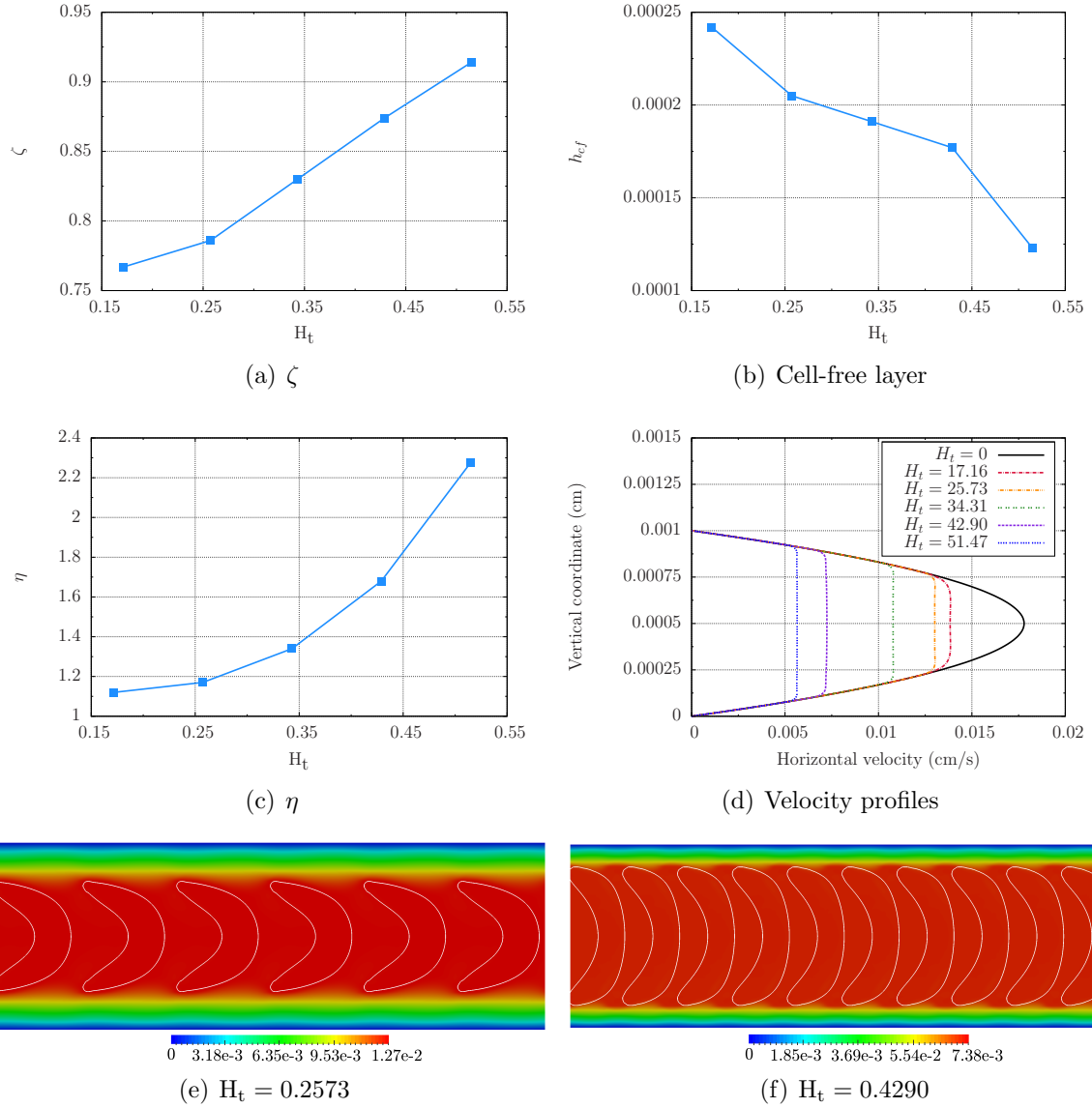
In this Section, we consider an off-center capsule and study its motion and shape evolution. The domain is the rectangle of 0.002×0.001 cm^2 drawn in Fig. 7 a), which is discretized using an Eulerian mesh with 256×128 cubic NURBS elements. The geometry of the capsule is a hollow ellipse with thickness $h = 5 \cdot 10^{-7}$ cm and semiaxes $a = 0.0002$ cm and $b = 0.0001$ cm. The initial position of the capsule is indicated in Fig. 7 a). The capsule is discretized using a Lagrangian mesh with 1×360 quadratic elements. The remaining physical parameters are the following: $\mu = 0.012$ g/(cm s), $\rho_1 = 1.025$ g/cm^3, $\boldsymbol{g} = (1666.66$ dyn/cm$^3, 0$ dyn/cm$^3)$, $\rho_2 = 1.1$ g/cm^3, and $G_m = 7500$ dyn/cm^2. The time step is $\Delta t = 0.00025$ s.

In a flow driven by a pressure gradient, vesicles and RBCs undergo a lateral migration towards the central region of the tube where the shear rate is minimal [67, 68, 69]. Figs. 7 b) to 7 f) describe this migration process for the considered capsule. During the lateral migration, the capsule varies its shape from a hollow ellipse to a parachute-like shape, which is often acquired by RBCs in small tubes [69]. Finally, we would like to mention that the membrane undergoes a tank-treading motion while the solution to the problem is non-symmetric. A small piece of the membrane is colored in black in order to show this motion in Figs. 7 b)-7 f). Once the capsule acquires a symmetric geometry, i.e., the parachute-like shape, the tank-treading motion stops.

4.3. Rheology of RBCs

In large arteries and veins, the concentration of RBCs acquires a nearly constant value with average 45%. However, in microcirculation, the concentration of RBCs varies strongly between different vessels due to the so-called plasma skimming effect [70]. In this Section, we consider different numbers of RBCs which are, in all cases, evenly distributed in the parabolic flow defined in Section 4.1. Our aim is to analyze important hemorheological properties such as, e.g., the Fåhræus and the Fåhræus-Lindqvist effects.

There are two different ways to measure the concentration of RBCs which are the tube hematocrit ($\mathrm{H_t}$) and the discharge hematocrit ($\mathrm{H_d}$). In a two-dimensional setting, $\mathrm{H_t}$ is the ratio of the area enclosed by the RBCs to the blood area and $\mathrm{H_d}$ is the ratio of the RBCs flow to the blood flow. In this Section we consider five cases: four RBCs ($\mathrm{H_t} = 0.1716$), six RBCs ($\mathrm{H_t} = 0.2573$), eight RBCs ($\mathrm{H_t} = 0.3431$), ten RBCs ($\mathrm{H_t} = 0.4290$), and twelve



(a) ζ (b) Cell-free layer

(c) η (d) Velocity profiles

(e) $H_t = 0.2573$ (f) $H_t = 0.4290$

Figure 8: *In silico* evaluation of hemorheological properties of blood. (a) Fåhræus effect. As the concentration of RBCs increases, the difference between the mean velocity of RBCs and the mean velocity of blood plasma diminishes. (b) Thickness of the cell-free layer with respect to the tube hematocrit. (c) Fåhræus-Lindqvist effect. The presence of RBCs increases the apparent viscosity of blood. (d) Blunt velocity profiles for different concentrations of RBCs. (e) Velocity magnitude and deformed RBC shapes for $H_t = 0.2573$. (f) Velocity magnitude and deformed RBC shapes for $H_t = 0.4290$.

RBCs ($\mathrm{H_t} = 0.5147$). In microcirculation, ($\mathrm{H_t}$) and ($\mathrm{H_d}$) are significantly different and this phenomenon is known as Fåhræus effect. Thus, we define ζ as

$$\zeta = \frac{\mathrm{H_t}}{\mathrm{H_d}} = \frac{v_{RBCs}}{v_m}, \tag{23}$$

where v_{RBCs} is the mean velocity of RBCs and v_m is the mean velocity of blood. It is known that ζ depends on the concentration of RBCs, the tube diameter (width in our two-dimensional setting), and the strength of the flow. Fig. 8 a) plots the values of ζ for the five different tube hematocrits considered. In accordance with the empirical laws obtained from *in vitro* and *in vivo* experiments [71, 72, 73], ζ increases with the value of $\mathrm{H_t}$. The Fåhræus effect is a consequence of the different mean velocities of RBCs and blood plasma. These two velocities are significantly different in small tubes due to the fact that RBCs tend to travel in the central region of the tube where the velocity is higher. Furthermore, a cell-free layer is known to develop in the boundary of the tube [74]. The thickness of the cell-free layer (h_{cf}) is plotted in Fig. 8 b) for the five different tube hematocrits considered.

In small tubes, the presence of RBCs affects the flow resistence for a certain pressure gradient, i.e., blood behaves as a non-Newtonian fluid where the viscosity is no longer constant. This phenomenon is known as Fåhræus-Lindqvist effect. We define η as the ratio of the blood flow, i.e., the flow considering both blood plasma and RBCs (Q) to the blood plasma flow, i.e., the flow of a simulation where only blood plasma is considered (Q_F). Thus,

$$\eta = \frac{Q_F}{Q} = \frac{v_F}{v_m}, \tag{24}$$

where v_F is the mean velocity of the flow when only blood plasma is considered in the simulation. The value of η depends on the concentration of RBCs, the tube diameter, and the strength of the flow. The parameter η is often called relative apparent viscosity since a Newtonian fluid with viscosity $\eta\mu$ would have the same mean velocity than blood for the particular pressure gradient and tube width considered. Fig. 8 c) plots the values of η for the five tube hematocrits considered. Consistently with empirical laws obtained from *in vitro* and *in vivo* experiments [71, 72, 73], η increases with the value of $\mathrm{H_t}$. We have also computed the velocity profiles once the RBCs have acquired a nearly constant shape. The results for blood plasma and the considered tube hematocrits are plotted in Fig. 8 d). As observed in [75], the velocity develops a blunt profile due to the presence of the RBCs. Finally, Figs. 8 e) and f) show the deformed RBC shapes at time $t = 1$ s for $\mathrm{H_t} = 0.2573$ and $\mathrm{H_t} = 0.4290$, respectively.

4.4. CTC passing through a channel narrowing

In this Section, we consider a CTC going through a narrowing. The computational domain is depicted in Fig. 9. The flow is driven by the forcing $\boldsymbol{g} = (1666.66\ \mathrm{dyn/cm^3}, 0\ \mathrm{dyn/cm^3})$. No-slip boundary conditions are applied in the vertical direction and periodic boundary conditions are applied in the horizontal direction. The CTC is modeled using a compound capsule. The membrane is a hollow circle with radius $r_m = 0.0005$ cm and thickness

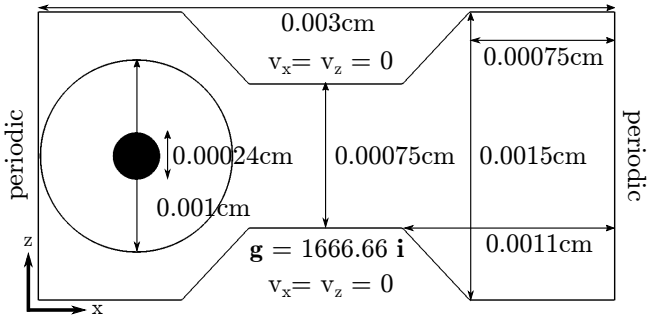


Figure 9: Circulating tumor cell going through a tube narrowing. Problem description of our two-dimensional setting.

$h = 5 \cdot 10^{-7}$ cm. The inclusion is a circle of radius $r_i = 0.00012$ cm. The remaining physical parameters are: $\mu = 0.012$ g/(cm s), $\rho_1 = 1.025$ g/cm^3, $\rho_2 = 1.1$ g/cm^3, $G_m = 7500$ dyn/cm^2, $G_i = 40$ dyn/cm^2.

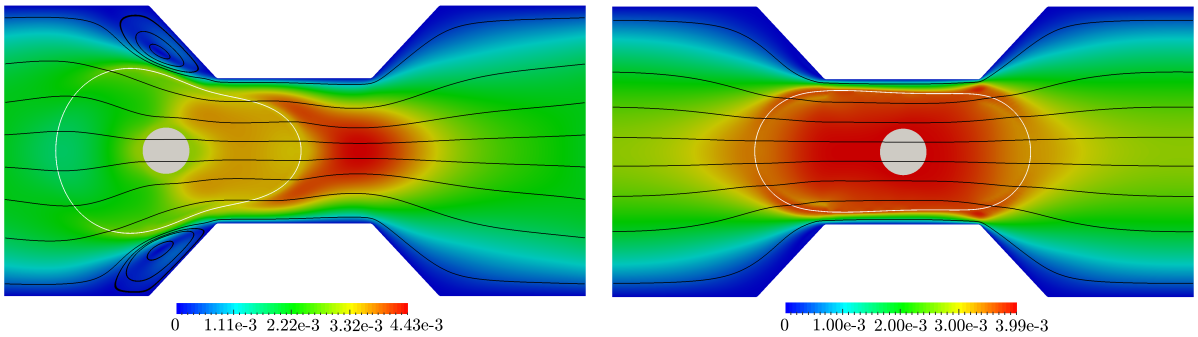
An Eulerian mesh with 384×192 cubic NURBS elements is used to represent the channel. Lagrangian meshes with 1×832 and 40×192 quadratic NURBS elements are used to represent the membrane and the inclusion, respectively. The time step used in the simulation is $\Delta t = 0.00025$ s.

Fig. 10 plots the velocity magnitude along with the streamlines at different time steps. These snapshots describe how the CTC deforms in order to go through the tube narrowing. The highest deformation of the cell and the highest velocity of the flow take place when the CTC is about to leave the narrowing; see Fig. 10 d). The CTC creates two recirculation regions when it is entering the narrowing as shown in Fig. 10 a). Recirculation regions appear as well when the CTC is leaving the narrowing; see Fig. 10 e). Due to the symmetry of the simulation, there is no TT motion of the membrane at any time. In Fig. 10 f), the CTC has already completed a loop in the periodic domain and is about to enter the narrowing for the second time.

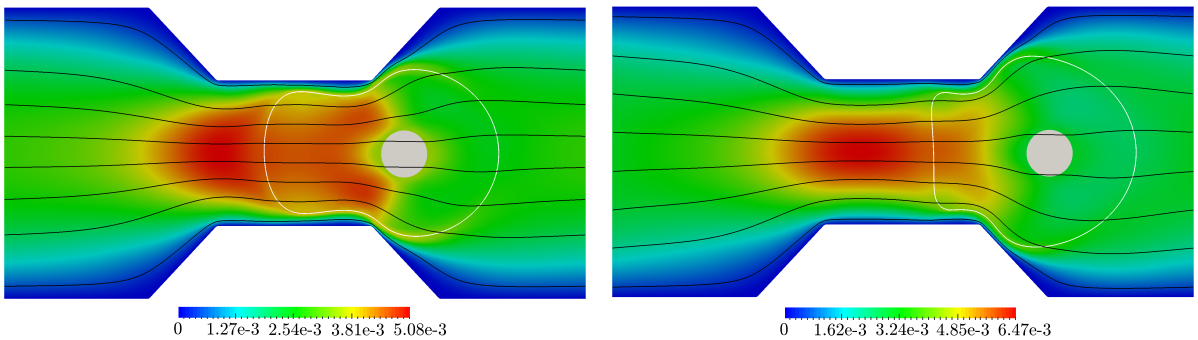
5. Conclusions and future work

This paper shows that our NURBS-based immersed FSI method is able to capture the behavior of capsules under different flow conditions as long as the incompressibility constraint is adequately satisfied close to the fluid-solid interface. In order to do so, a parameter sweep needs to be performed in advance so as to find a good balance between constraint accuracy, convergence of the solution and computational efficiency. Our results, which are in agreement with experimental and numerical works [16, 69, 72, 73, 74, 75], show that capsules, modeled as solid-shell NURBS elements, are good numerical proxies for red blood cells.

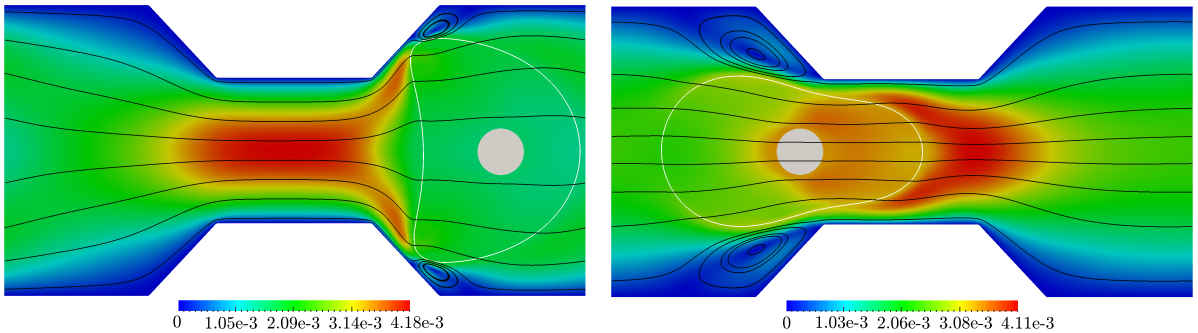
Inspired by the compound vesicle model presented in [16], we propose hyperelastic compound capsules as a numerical proxy for nucleated cells embedded in flow, such as, e.g., WBCs and CTCs. Our results for the dependence of the inclination angle and tank-



(a) $t = 0.1$ s (b) $t = 0.3$ s



(c) $t = 0.4375$ s (d) $t = 0.5$ s



(e) $t = 0.625$ s (f) $t = 1.075$ s

Figure 10: Snapshots of the velocity magnitude along with the streamlines for a CTC going through a tube narrowing. Note the different scale for each time. Both the highest flow velocity and deformation of the membrane appear when the CTC is about to leave the narrowing. Recirculation regions appear when the CTC is entering and leaving the narrowing as shown in a) and e), respectively.

treading frequency with the filling fraction are in agreement with results for compound vesicles [16]. Additionally, our geometrically-flexible FSI approach enables us to consider a channel narrowing and analyze how a CTC passes through it. A deeper understanding of this process may provide ideas to improve existent techniques of CTC sorting [76].

Although we have focused on WBCs and CTCs, hyperelastic compound capsules can also be of interest to predict the behavior of composite core-shell microparticles, which can in turn help to understand the mechanical behaviour of analagous biomaterials, such as liposomes [77]. The fact that a compound capsule accounts for mechanical strains of the shell and deformability of the nucleus is a first step to better characterize the mechanical behavior of cells. Recent developments in microfluidics have shown the importance of cell deformability as a biomarker, allowing for metastatic cancer cells to be separated from blood cells, benign cancer cells and normal tissue cells from the same origin by means of inertial fluid effects [78]. In this regard, we believe that the combination of our immersed FSI method, which does not neglect inertial effects, with compound capsules looks highly promising to model these systems, even though more rigorous validation is required. Further improvements of this work include extending the FSI model to be able to handle different viscosities for the fluid inside and outside the membrane, comparing different material models for the membrane, and studying the interaction between RBCs and CTCs in microcirculation.

Acknowledgements

HG, HC and CB-C were partially supported by the European Research Council through the FP7 Ideas Starting Grant (project # 307201). HG was also partially supported by Xunta de Galicia, and by Ministerio de Economía y Competitividad (contract # DPI2013-44406-R). The last two grants are cofinanced with FEDER funds. This support is gratefully acknowledged. We acknowledge the open source scientific libraries PETSc, PetIGA, and their developers.

References

[1] R. Fåhræus, T. Lindqvist, The viscosity of the blood in narrow capillary tubes, American Journal of Physiology–Legacy Content 96 (3) (1931) 562–568.

[2] T. Fischer, M. Stohr-Lissen, H. Schmid-Schonbein, The red cell as a fluid droplet: tank tread-like motion of the human erythrocyte membrane in shear flow, Science 202 (1978) 894–896.

[3] Y. C. Fung, Biomechanics, Springer, New York, 1990.

[4] J. B. Freund, Numerical simulation of flowing blood cells, Annual review of fluid mechanics 46 (2014) 67–95.

[5] D. Abreu, M. Levant, V. Steinberg, U. Seifert, Fluid vesicles in flow, Advances in colloid and interface science 208 (2014) 129–141.

[6] S. K. Veerapaneni, D. Gueyffier, D. Zorin, G. Biros, A boundary integral method for simulating the dynamics of inextensible vesicles suspended in a viscous fluid in 2D, Journal of Computational Physics 228 (7) (2009) 2334 – 2353.

[7] S. K. Veerapaneni, D. Gueyffier, G. Biros, D. Zorin, A numerical method for simulating the dynamics of 3D axisymmetric vesicles suspended in viscous flows, Journal of Computational Physics 228 (19) (2009) 7233 – 7249.

[8] C. Pozrikidis, Modeling and simulation of capsules and biological cells, CRC Press, 2003.

[9] G. Késmárky, P. Kenyeres, M. Rábai, K. Tóth, Plasma viscosity: a forgotten variable, Clinical hemorheology and microcirculation 39 (1–4) (2008) 243–246.

[10] H. W. Hou, M. E. Warkiani, B. L. Khoo, Z. R. Li, R. A. Soo, D. S.-W. Tan, W.-T. Lim, J. Han, A. A. S. Bhagat, C. T. Lim, Isolation and retrieval of circulating tumor cells using centrifugal forces, Scientific Reports 3 (2013) 1259.

[11] M. E. Warkiani, B. L. Khoo, L. Wu, A. K. P. Tay, A. A. S. Bhagat, J. Han, C. T. Lim, Ultra-fast, label-free isolation of circulating tumor cells from blood using spiral microfluidics, Nat. Protocols 11 (1) (2016) 134–148.

[12] T. J. R. Hughes, G. Feijóo, L. Mazzei, J.-B. Quincy, The variational multiscale method - A paradigm for computational mechanics, Computer Methods in Applied Mechanics and Engineering 166 (1998) 3–24.

[13] T. J. R. Hughes, G. Sangalli, Variational multiscale analysis: The fine-scale Green's function, projection, optimization, localization, and stabilized methods, SIAM Journal on Numerical Analysis 45 (2007) 539–557.

[14] Y. Bazilevs, V. M. Calo, J. A. Cottrell., T. J. R. Hughes, A. Reali, G. Scovazzi, Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows, Computer Methods in Applied Mechanics and Engineering 197 (2007) 173–201.

[15] R. Tran-Son-Tay, G. B. Nash, Hemorheology and hemodynamics, IOS Press, Amsterdam, 2007.

[16] S. K. Veerapaneni, Y.-N. Young, P. M. Vlahovska, J. Bławzdziewicz, Dynamics of a compound vesicle in shear flow, Physical review letters 106 (15) (2011) 158103.

[17] P. Koumoutsakos, I. Pivkin, F. Milde, The fluid mechanics of cancer and its therapy, Annual Review of Fluid Mechanics 45 (2013) 325–355.

[18] Y. Liu, L. Zhang, X. Wang, W. K. Liu, Coupling of Navier-Stokes equations with protein molecular dynamics and its application to hemodynamics, International Journal for Numerical Methods in Fluids 46 (2004) 1237–1252.

[19] Y. Liu, W. K. Liu, Rheology of red blood cell aggregation by computer simulation, Journal of Computational Physics 220 (2006) 139–154.

[20] W. K. Liu, S. Jun, Y. F. Zhang, Reproducing kernel particle methods, International Journal for Numerical Methods in Fluids 20 (8-9) (1995) 1081–1106.

[21] W. K. Liu, S. Jun, S. Li, J. Adee, T. Belytschko, Reproducing kernel particle methods for structural dynamics, International Journal for Numerical Methods in Engineering 38 (1995) 1655–1679.

[22] S. Lipton, J. Evans, Y. Bazilevs, T. Elguedj, T. J. R. Hughes, Robustness of isogeometric structural discretizations under severe mesh distortion, Computer Methods in Applied Mechanics and Engineering 199 (2010) 357–373.

[23] H. Casquero, C. Bona-Casas, H. Gomez, A NURBS-based immersed methodology for fluid-structure interaction, Computer Methods in Applied Mechanics and Engineering 284 (2015) 943–970.

[24] H. Casquero, L. Liu, C. Bona-Casas, Y. Zhang, H. Gomez, A hybrid variational-collocation immersed method for fluid-structure interaction using unstructured t-splines, International Journal for Numerical Methods in Engineering 105 (11) (2016) 855–880.

[25] F. Auricchio, L. Beirao Da Veiga, T. J. R. Hughes, A. Reali, G. Sangalli, Isogeometric collocation methods, Mathematical Models and Methods in Applied Sciences 20 (2010) 2075–2107.

[26] H. Casquero, L. Liu, Y. Zhang, A. Reali, H. Gomez, Isogeometric collocation using analysis-suitable T-splines of arbitrary degree, Computer Methods in Applied Mechanics and Engineering 301 (2016) 164 – 186.

[27] D. Schillinger, J. Evans, A. Reali, M. Scott, T. J. R. Hughes, Isogeometric collocation: Cost comparison with Galerkin methods and extension to adaptive hierarchical NURBS discretizations, Computer Methods in Applied Mechanics and Engineering 267 (2013) 170–232.

[28] H. Gomez, A. Reali, G. Sangalli, Accurate, efficient, and (iso)geometrically flexible collocation methods for phase-field models, Journal of Computational Physics 262 (2014) 153–171.

[29] I. Babuška, Error-bounds for finite element method, Numerische Mathematik 16 (4) (1971) 322–333.

[30] F. Brezzi, On the existence, uniqueness and approximation of saddle-point problems arising from lagrangian multipliers, ESAIM: Mathematical Modelling and Numerical Analysis 8 (1974) 129–151.

[31] H. Gomez, L. De Lorenzis, The variational collocation method, Computer Methods in Applied Mechanics and Engineering, DOI: 10.1016/j.cma.2016.06.003.

[32] D. Kamensky, M.-C. Hsu, D. Schillinger, J. A. Evans, A. Aggarwal, Y. Bazilevs, M. S. Sacks, T. J. R. Hughes, An immersogeometric variational framework for fluid-structure interaction: Application to bioprosthetic heart valves, Computer Methods in Applied Mechanics and Engineering 284 (2015) 1005–1053.

[33] J. A. Cottrell, T. J. R. Hughes, Y. Bazilevs, Isogeometric Analysis Toward Integration of CAD and FEA, Wiley, 2009.

[34] M.-C. Hsu, D. Kamensky, Y. Bazilevs, M. Sacks, T. J. R. Hughes, Fluid-structure interaction analysis of bioprosthetic heart valves: Significance of arterial wall deformation, Computational Mechanics 54 (2014) 1055–1071.

[35] M.-C. Hsu, D. Kamensky, F. Xu, J. Kiendl, C. Wang, M. Wu, J. Mineroff, A. Reali, Y. Bazilevs, M. Sacks, Dynamic and fluid-structure interaction simulations of bioprosthetic heart valves using parametric design with T-splines and Fung-type material models, Computational Mechanics 55 (6) (2015) 1211–1225.

[36] L. Zhang, A. Gerstenberger, X. Wang, W. K. Liu, Immersed finite element method, Computer Methods in Applied Mechanics and Engineering 193 (2004) 2051–2067.

[37] W. K. Liu, Y. Liu, D. Farrell, L. Zhang, X. Wang, Y. Fukui, N. Patankar, Y. Zhang, C. Bajaj, J. Lee, J. Hong, X. Chen, H. Hsu, Immersed finite element method and its applications to biological systems, Computer Methods in Applied Mechanics and Engineering 195 (2006) 1722–1749.

[38] K. Jansen, C. Whiting, G. Hulbert, Generalized-α method for integrating the filtered navier-stokes equations with a stabilized finite element method, Computer Methods in Applied Mechanics and Engineering 190 (2000) 305–319.

[39] T. E. Tezduyar, S. Sathe, Modelling of fluid-structure interactions with the space-time finite elements: Solution techniques, International Journal for Numerical Methods in Fluids 54 (6-8) (2007) 855–900.

[40] Y. Bazilevs, K. Takizawa, T. E. Tezduyar, Computational fluid-structure interaction: methods and applications, John Wiley & Sons, 2012.

[41] P. R. Brune, M. G. Knepley, B. F. Smith, X. Tu, Composing scalable nonlinear algebraic solvers, SIAM Review 57 (4) (2015) 535–565.

[42] Y. Saad, M. H. Schultz, Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM Journal on scientific and statistical computing 7 (1986) 856–869.

[43] C. Michler, E. van Brummelen, R. de Borst, Error-amplification analysis of subiteration-preconditioned gmres for fluid-structure interaction, Computer Methods in Applied Mechanics and Engineering 195 (17-18) (2006) 2124–2148.

[44] L. Dalcin, N. Collier, P. Vignal, A. Crtes, V. Calo, PetIGA: A framework for high-performance isogeometric analysis, Computer Methods in Applied Mechanics and Engineering 308 (2016) 151 – 181.

[45] P. A. Vignal, N. Collier, V. M. Calo, Phase field modeling using PetIGA, Procedia Computer Science 18 (2013) 1614–1623.

[46] M. Bernal, V. M. Calo, N. Collier, G. Espinosa, F. Fuentes, J. Mahecha, Isogeometric analysis of hyperelastic materials using PetIGA, Procedia Computer Science 18 (2013) 1604–1613.

[47] S. Balay, M. F. Adams, J. Brown, P. Brune, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, H. Zhang, PETSc Web page, `http://www.mcs.anl.gov/petsc` (2014).

[48] S. Balay, M. F. Adams, J. Brown, P. Brune, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, H. Zhang, PETSc users manual, Argonne National Laboratory (ANL-95/11 - Revision 3.4).
URL `http://www.mcs.anl.gov/petsc`

[49] K. Singh, J. Williams, Application of the additive schwarz method to large scale poisson problems, Communications in numerical methods in engineering 20 (3) (2004) 193–205.

[50] H. Schmid-Schönbein, R. Wells, Fluid drop-like transition of erythrocytes under shear, Science 165 (3890) (1969) 288–291.

[51] H. Goldsmith, J. Marlow, Flow behaviour of erythrocytes. i. rotation and deformation in dilute suspensions, Proceedings of the Royal Society of London B: Biological Sciences 182 (1068) (1972) 351–384.

[52] H. Basu, A. K. Dharmadhikari, J. A. Dharmadhikari, S. Sharma, D. Mathur, Tank treading of optically trapped red blood cells in shear flow, Biophysical journal 101 (7) (2011) 1604–1612.

[53] D. A. Fedosov, M. Peltomäki, G. Gompper, Deformation and dynamics of red blood cells in flow through cylindrical microchannels, Soft Matter 10 (24) (2014) 4258–4267.

[54] R. Skalak, N. Ozkaya, , T. C. Skalak, Biofluid mechanics, Annual Review of Fluid Mechanics 21 (1) (1989) 167–200.

[55] R. Skalak, A. Tozeren, R. Zarda, S. Chien, Strain energy function of red blood cell membranes, Biophysical Journal 13 (3) (1973) 245 – 264.

[56] P. Bagchi, R. M. Kalluri, Dynamics of nonspherical capsules in shear flow, Physical Review E 80 (1) (2009) 016307.

[57] Y. Sui, Y. Chew, P. Roy, H. Low, A hybrid method to study flow-induced deformation of three-dimensional capsules, Journal of Computational Physics 227 (12) (2008) 6351 – 6371.

[58] S. Ramanujan, C. Pozrikidis, Deformation of liquid capsules enclosed by elastic membranes in simple shear flow: large deformations and the effect of fluid viscosities, Journal of Fluid Mechanics 361 (1998) 117–143.

[59] T. J. R. Hughes, J. A. Cottrell, Y. Bazilevs, Isogeometric analysis CAD, finite elements, NURBS, exact geometry and mesh refinement, Computacional Methods in Applied Mechanics and Engineering 194 (2005) 4135–4195.

[60] B. Kaoui, J. Harting, C. Misbah, Two-dimensional vesicle dynamics under shear flow: Effect of confinement, Physical Review E 83 (6) (2011) 066319.

[61] M. R. King, K. G. Phillips, A. Mitrugno, T.-R. Lee, A. M. de Guillebon, S. Chandrasekaran, M. J. McGuire, R. T. Carr, S. M. Baker-Groberg, R. A. Rigg, et al., A physical sciences network characterization of circulating tumor cell aggregate transport, American Journal of Physiology-Cell Physiology 308 (10) (2015) C792–C802.

[62] I. Cantat, C. Misbah, Lift force and dynamical unbinding of adhering vesicles under shear flow, Physical review letters 83 (4) (1999) 880.

[63] D. A. Fedosov, J. Fornleitner, G. Gompper, Margination of white blood cells in microcapillary flow, Physical review letters 108 (2) (2012) 028104.

[64] O. K. Baskurt, Handbook of hemorheology and hemodynamics, Vol. 69, IOS press, 2007.

[65] B. Kaoui, T. Krüger, J. Harting, Complex dynamics of a bilamellar vesicle as a simple model for leukocytes, Soft Matter 9 (33) (2013) 8057–8061.

[66] Z. Y. Luo, L. He, B. F. Bai, Deformation of spherical compound capsules in simple shear flow, Journal of Fluid Mechanics 775 (2015) 77–104.

[67] G. Coupier, B. Kaoui, T. Podgorski, C. Misbah, Noninertial lateral migration of vesicles in bounded poiseuille flow, Physics of Fluids (1994-present) 20 (11) (2008) 111702.

[68] G. Danker, P. M. Vlahovska, C. Misbah, Vesicles in poiseuille flow, Physical review letters 102 (14) (2009) 148102.

[69] J. L. McWhirter, H. Noguchi, G. Gompper, Deformation and clustering of red blood cells in microcapillary flows, Soft Matter 7 (22) (2011) 10967–10977.

[70] A. Krogh, The anatomy and physiology of capillaries, Vol. 18, Yale University Press, 1922.

[71] R. Fåhraeus, The suspension stability of the blood, Physiological Reviews 9 (2) (1929) 241–274.

[72] S. Chien, S. Usami, R. Skalak, Blood flow in small tubes, Handbook of physiology, section 2 (1984) 217–249.

[73] A. R. Pries, T. W. Secomb, Microvascular blood viscosity in vivo and the endothelial surface layer, American Journal of Physiology-Heart and Circulatory Physiology 289 (6) (2005) H2657–H2664.

[74] P. Olla, Simplified model for red cell dynamics in small blood vessels, Physical review letters 82 (2) (1999) 453.

[75] P. Bagchi, Mesoscale simulation of blood flow in small vessels, Biophysical journal 92 (6) (2007) 1858–1877.

[76] Z. Zhang, J. Xu, B. Hong, X. Chen, The effects of 3d channel geometry on ctc passing pressure - towards deformability-based cancer cell separation, Lab Chip 14 (2014) 2576–2584.

[77] T. Kong, L. Wang, H. M. Wyss, H. C. Shum, Capillary micromechanics for core-shell particles, Soft Matter 10 (2014) 3271–3276.

[78] S. C. Hur, N. K. Henderson-MacLennan, E. R. B. McCabe, D. Di Carlo, Deformability-based cell classification and enrichment using inertial microfluidics, Lab Chip 11 (2011) 912–920.

E Extended summary in Spanish

E.1 Introducción

En las últimas décadas, el desarrollo de metodologías de modelación y simulación para problemas de interacción fluido-estructura ha atraído enorme atención en la comunidad de la mecánica computacional. Un método computacional preciso, geométricamente flexible, robusto, y eficiente es crucial a la hora de resolver complicados problemas de interacción fluido-estructura, los cuales están presentes en ingeniería biomédica [15, 16], ingeniería aeroespacial [116, 117], ingeniería civil [86, 118, 98, 45], ingeniería marina [130, 127] e ingeniería eólica [12, 13] por nombrar unos ejemplos. Hasta la fecha, no hay un método de interacción fluido-estructura que pueda ser utilizado en todos los tipos de problemas de interacción fluido-estructura. Por lo tanto, diferentes tipos de métodos de interacciń fluido-estructura han sido desarrollados para aplicaciones de interacción fluido-estructura con distintas características.

Los métodos de interacción fluido-estructura basados en mallas pueden ser clasificados en dos grupos principales, en concreto, métodos ajustados al contorno y métodos inmersos. En los métodos ajustados al contorno, una malla es ajustada al dominio del fluido utilizando la descripción arbitraria de Euler-Lagrange y otra malla es ajustada al dominio del sólido utilizando la descripción Lagrangiana, estas dos mallas normalmente son conformes en la interfaz fluido-estructura de manera que la imposición de condiciones cinemáticas pueda ser realizada de forma exacta. La malla del fluido tiene que ser actualizada en cada paso de tiempo resolviendo un problema de valores de contorno donde las condiciones de Dirichlet aplicadas son los desplazamientos del sólido en la interfaz. A medida que el sólido se mueve, la calidad de la malla del fluido disminuirá y acabará llegando un momento donde el fluido necesitará ser remallado. Para una explicación minuciosa de los métodos ajustados al contorno y ejemplos de aplicación, el lector interesado puede consultar [17]. En métodos inmersos, una malla Euleriana abarca todo el dominio computacional y una malla Lagrangiana ees ajustada al dominio del sólido. Por lo tanto, la necesidad de remallar la malla ajustada al dominio del fluido ha desaparecido. Sin embargo, se obtiene esta ventaja al precio de haber asumido que el campo de la presión es continuo en la interfaz sólido-fluido y esto es especialmente problemático cuando se consideran sólidos muy delgados [67].

El movimiento del sólido es el factor clave para decidir si utilizar un método ajustado al contorno o un método inmerso. Si utilizamos un método ajustado al contorno en una aplicación donde el sólido experimenta grandes traslaciones o rotaciones, la malla ajustada al dominio del fluido se distorsionará mucho cada pocos pasos de tiempo, afectando la precisión de la simulación. El remallado puede ser utilizado para combatir este problema, pero, además de su alto coste computacional, se necesitarán proyecciones de la velocidad y de la presión entre la antigua y la nueva malla, lo cual introduce imprecisiones en la simulación. Estas dificultades se amplifican cuando hay varios sólidos y estos sólidos se mueven por todo el dominio, por ejemplo, en flujos de partículas [52]. Además, en casos donde la topología de la región ocupada por el fluido va a cambiar durante la simulación debido al contacto entre diferentes sólidos, la aplicación de un método ajustado al contorno es todavía un problema abierto. En todas las situaciones mencionadas anteriormente,

los métodos inmersos emergen como una alternativa adecuada a los métodos ajustados al contorno puesto que las mallas Lagrangianas asociadas a cada sólido pueden moverse libremente por encima de la malla Euleriana.

Los dos principales objetivos de esta tesis son el desarrollo de métodos inmersos para interacción fluido-estructura usando splines y aplicar dichos métodos al flujo sanguíneo a escala celular, el cual es uno de los flujos de partículas que ha recientemente recibido más atención debido a su conexión con diferentes enfermedades como el cáncer [77]. Funciones spline de combinación desarrolladas en el campo del diseño asistido por ordenador tales como, por ejemplo, NURBS [106] y T-splines [113], fueron utilizadas por primera vez en mecánica computacional en [63] y [10], respectivamente. La alta continuidad entre elementos, la no negatividad y flexibilidad geométrica de las funciones spline han originado importantes avances en tanto mecánica de fluidos [2, 11] como mecánica de sólidos [38, 81]. Por lo tanto, la aplicación de splines a métodos inmersos de interacción fluido-estructura basados en mallas parece una forma interesante de mejorar la precisión, la robustez y la flexibilidad geométrica de las versiones actuales de dichos métodos.

Esta tesis extiende el estado del arte en métodos inmersos de interacción fluido-estructura basados en mallas desarrollando los avances que se resumen a continuación:

1. **Un método inmerso basado en NURBS para interacción fluido-estructura**. Presentamos una formulación híbrida que combina un enfoque variacional con un enfoque de colocación, inmersa y completamente implícita para interacción fluido-estructura. Esta tócnica trabaja con fluidos que son viscosos e incompresibles y con sólidos que son no lineales e incompresibles. El análisis isogeométrico basado en NURBS se utiliza para la discretización espacial y el método alfa-generalizado se utiliza para la discretización temporal. La transferencia de datos entre la malla Euleriana y las mallas Lagrangiannas es llevada a cabo usando también las funciones NURBS, lo cual evita el uso de las denominadas deltas de Dirac discretizadas a menudo utilizadas en métodos inmersos. Validamos el método comparando nuestras soluciones numéricas con datos teóricos para velocidades de cuerpos rígidos en caída libre en dos y tres dimensiones. Ver la Sección 4 y el Apéndice A para detalles adicionales.

2. **Colocación isogeométrica basada en T-splines adecuadas para análisis de orden arbitrario**. Con el objetivo de extender nuestro método inmerso de interacción fluido-estructura a T-splines, hemos tenido que desarrollar primero métodos de colocación isogeométrica basados en T-splines adecuadas para análisis de orden arbitrario, lo cual representaba un problema abierto en mecánica computacional. T-splines adecuadas para análisis permiten realizar refinamiento de tipo h al mismo tiempo que mantienen tanto el coste de sólo una evaluación puntual por grado de libertad como los índices de convergencia de los métodos de colocacién basados en NURBS. Además, T-splines adecuados para análisis (a partir de ahora denominadas ASTS por sus iniciales en inglés) son capaces de crear mallas altamente no uniformes sin introducir elementos con cocientes de forma altos, lo cual evita

inestabilidades numéricas que aparecen en problemas con regularidad reducida cuando condiciones de contorno de Neumann son impuestas en forma fuerte en mallas con elementos con cocientes de forma altos. Ver la Sección 5 y el Apéndice B para detalles adicionales.

3. **Un método inmerso basado en ASTS para interacción fluido-estructura**. Nuestros métodos inmersos para interacción fluido-estructura son generalizados para trabajar con ASTS. Validamos nuestros resultados numéricos comparándolos con soluciones teóricas. ASTS poseen dos ventajas principales cuando son comparadas con NURBS, en concreto, ASTS permiten realizar refinamiento local de tipo h y son desestructuradas. Estas propiedades tienen el potencial de aumentar la eficiencia y flexibilidad de discretizaciones espaciales basadas en NURBS. Ver la Sección 6 y el Apéndice C para detalles adicionales.

4. **Cápsulas y cápsulas compuestas como imitadores celulares en flujo**. Realizamos simulaciones con cápsulas en flujos de cortes y flujos parabólicos con el objectivo de reproducir los principales movimientos y formas que adquieren los glóbulos rojos en estos flujos. También medidos y comparamos con experimentos varias propiedades reológicas de la sangre en pequeñas escalas. Además, presentamos el concepto de cápsula compuesta como un imitador celular para células con núcleo tales como glóbulos blancos y células cancerosas en circulación. Realizamos simulaciones con cápsulas compuestas hiperelásticas en flujo de corte con el objetivo de estudiar su comportamiento en dos y tres dimensiones. Finalmente, mostramos la habilidad de las células cancerosas en circulación para atravesar un estrechamiento de canal, esta propiedad está actualmente siendo utilizada para separar células cancerosas en circulación en muestras de sangre [134]. Ver la Sección 7 y el Apéndice C para detalles adicionales.

E.2 Objetivos

Los objetivos de esta tesis pueden ser clasificados en tres categoras:

- Desarrollo y validación de métodos numéricos. El objetivo es idear métodos inmersos para interacción fluido-estructura basados en NURBS y ASTS. Dichos métodos deberán aprovecharse de la alta continuidad entre elementos de las funciones base spline con el fin de aumentar la precisión y robustez con respecto a anteriores métodos inmersos como el método de elementos finitos inmersos [131]. Además, el uso de NURBS y ASTS permitiría la representación exacta de curvas cónicas y superficies cuadráticas que aparecen a menudo en problemas de interacción fluido-estructura. Los métodos propuestos deben ser validados resolviendo problemas con solución teórica conocida.

- Aplicación a biomecánica. El objetivo es estudiar el comportamiento de diferentes tipos de células sanguíneas en flujo sanguíneo en pequeñas escalas. Varios autores han mencionado que los polinomios de Lagrange no son capaces de manejar las grandes deformaciones de los glóbulos rojos en

flujo de corte y flujo parabólico [90, 91], proponiendo métodos alternativos basados en partículas [88, 87]. La mayor robustez de las funciones NURBS es utilizada para superar este problema. Con el fin de evaluar los resultados de nuestras simulaciones, éstos deben ser comparados con experimentos de glóbulos rojos. Además, pretendemos desarrollar modelos para células con núcleo como los glóbulos blancos y las células cancerosas en circulación. Finalmente, ejemplos en geometrías no triviales deberían ser considerados para ilustrar la flexibilidad geométrica de nuestros métodos inmersos.

- Implementación de los algoritmos propuestos. La mayoría de los métodos inmersos para interacción fluido-estructura basados en mallas han sido sólo implementados en códigos seriales. En esta tesis, el objetivo es ir un paso más allá y realizar implementaciones en paralelo de nuestros algoritmos.

E.3 Definición del problema a resolver

A continuación se procede a enunciar el problema matemático a resolver en esta tesis. Sea $d = \{2, 3\}$ el número de dimensiones espaciales. Sea Ω_t^f y $\Omega_t^s \subset \mathbb{R}^d$ dos conjuntos abiertos que representan los dominios ocupados por el fluido viscoso incompresible y el sólido hiperelástico incompresible, respectivamente, con Γ_t^f y Γ_t^s sus correspondientes contornos. Estas dos regiones tienen parte de su contorno en común, a este contorno lo denominaremos interfaz fluido-s´lido Γ_t^I ($\Gamma_t^I = \overline{\Omega_t^f} \cap \overline{\Omega_t^s}$). Sea $\Omega \subset \mathbb{R}^d$ un conjunto abierto que representa el dominio ocupado por tanto el fluido como el sólido, es decir, $\overline{\Omega} = \overline{\Omega_t^f \cup \Omega_t^s}$. Sea Γ el contorno de Ω. Sean $\boldsymbol{v} : \Omega \times (0, T) \mapsto \mathbb{R}^d$, $p : \Omega \times (0, T) \mapsto \mathbb{R}$ y $\boldsymbol{u} : \Omega_0^s \times (0, T) \mapsto \mathbb{R}^d$ la velocidad Euleriana, la presión Euleriana y el desplazamiento Lagrangiano, respectivamente, donde $(0, T)$ es el intervalo temporal de interés y Ω_0^s es la configuración de referencia de Ω_t^s. Sean $\boldsymbol{X} \in \Omega_0^s$, $\boldsymbol{x} \in \Omega$, $\boldsymbol{x} \in \Omega$ y $\boldsymbol{\varphi} : \Omega_0^s \times (0, T) \mapsto \Omega_t^s$ un punto material, un punto espacial, y el mapeo $\boldsymbol{\varphi}(\boldsymbol{X}, t) = \boldsymbol{X} + \boldsymbol{u}(\boldsymbol{X}, t)$, respectivamente.

Las tres ecuaciones que definen nuestro problema son la ecuación de balance de momento lineal, la ecuación de conservación de masa y la relación cinemática entre el desplazamiento Lagrangiano y la velocidad Euleriana:

$$\rho^f \left(\frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v} \cdot \nabla_{\boldsymbol{x}} \boldsymbol{v} \right) = \nabla_{\boldsymbol{x}} \cdot \boldsymbol{\sigma}^f + \rho^f \boldsymbol{g} + \boldsymbol{\mathcal{F}} \qquad \text{in} \quad \Omega \times (0, T) \,, \tag{38}$$

$$\nabla_{\boldsymbol{x}} \cdot \boldsymbol{v} = 0 \qquad \text{in} \quad \Omega \times (0, T) \,, \tag{39}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} = \boldsymbol{v} \qquad \text{in} \quad \Omega_t^s \times (0, T) \,, \tag{40}$$

con:

$$\boldsymbol{\mathcal{F}} = \begin{cases} 0, & \boldsymbol{x} \in \Omega_t^f \\ \left(\rho^f - \rho^s\right)\left(\frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v} \cdot \nabla_{\boldsymbol{x}} \boldsymbol{v} - \boldsymbol{g}\right) + \nabla_{\boldsymbol{x}} \cdot \left(\boldsymbol{\sigma}^s - \boldsymbol{\sigma}^f\right), & \boldsymbol{x} \in \Omega_t^s \end{cases} , \tag{41}$$

$$\boldsymbol{\sigma}^f = -p\boldsymbol{I} + 2\mu \nabla_{\boldsymbol{x}}^{\text{sym}} \boldsymbol{v} \,, \tag{42}$$

$$\boldsymbol{\sigma}^s = -p\boldsymbol{I} + \boldsymbol{F}\boldsymbol{S}\boldsymbol{F}^T / J \,, \tag{43}$$

donde ρ^f y ρ^s son la densidad del fluido y del sólido, respectivamente, $\boldsymbol{\sigma}^f$ y $\boldsymbol{\sigma}^s$ son el tensor de tensiones de Cauchy del fluido y del sólido, respectivamente, $\boldsymbol{g}$ es una fuerza de volumen externa actuando en el sistema, $\boldsymbol{I}$ denota el tensor identidad en $\mathbb{R}^{d\times d}$, μ is la viscosidad dinámica del fluido, $\nabla_{\boldsymbol{x}}^{\text{sym}}(\cdot)$ es el operador gradiente simétrico definido como $\nabla_{\boldsymbol{x}}^{\text{sym}}\boldsymbol{v} = (\nabla_{\boldsymbol{x}}\boldsymbol{v} + \nabla_{\boldsymbol{x}}\boldsymbol{v}^T)/2$, $\boldsymbol{F} : \Omega_0^s \times (0,T) \mapsto \mathbb{R}^{d\times d}$ es el gradiente de deformación definido como, $\boldsymbol{F} = \nabla_{\boldsymbol{X}}\boldsymbol{\varphi} = \boldsymbol{I} + \nabla_{\boldsymbol{X}}\boldsymbol{u}$, $\boldsymbol{S}$ es el segundo tensor de tensiones de Piola-Kirchhoff del material considerado y $J = \det(\boldsymbol{F})$ es el determinante del Jacobiano el cual es exactamente igual a 1 a nivel continuo ya que el sólido es incompresible.

Adecuadas condiciones de contorno e iniciales deben ser impuestas a las ecuaciones anteriores para obtener un problema de interacción fluido-estructura bien definido. Con el objetivo de imponer condiciones de contorno, dividimos el contorno Γ en dos conjunto $\mathcal{B}^s = \Gamma \cap \Gamma_t^s$ y $\mathcal{B}^f = \Gamma \cap \Gamma_t^f$ de manera que $\Gamma = \overline{\mathcal{B}^s \cup \mathcal{B}^f}$ y $\emptyset = \mathcal{B}^s \cap \mathcal{B}^f$. Es importante darse cuenta que $\mathcal{B}^s \neq \Gamma_t^s$ puesto que por lo menos parte del sólido está inmerso en el fluido. En la interfaz Γ_t^I, dos condiciones de contorno son impuestas, en concreto, igualdad de velocidades y compatibilidad de tensiones entre el fluido y el sólido. La igualdad de velocidades se impone por medio de la ecuación (40) y la compatibilidad de tensiones se impone en la forma débil de la ecuación (38). En $\mathcal{B}^f$, distinguimos entre condiciones de contorno de Dirichlet en $\mathcal{B}_D^f$ y condiciones de contorno de Neumann en $\mathcal{B}_N^f$ de manera que $\mathcal{B}^f = \overline{\mathcal{B}_D^f \cup \mathcal{B}_N^f}$ y $\mathcal{B}_D^f \cap \mathcal{B}_N^f = \emptyset$. En $\mathcal{B}^s$, consideramos condiciones de contorno Dirichlet homogéneas. Con respecto a las condiciones iniciales, la posición inicial del sólido, el desplazamiento Lagrangiano y la velocidad Euleriana deben ser especificadas.

E.4 Contribuciones de la tesis

Esta sección resume las principales contribuciones e ideas de esta tesis. Hemos desarrollado métodos inmersos basados en mallas para interacción fluido-estructura con las siguientes características:

- Se asume que tanto el fluido como el sólido son incompresibles, lo cual es un caso de especial interés en varios problemas de biomecánica [50]. Además de eso, trabajamos con fluidos viscosos Newtonianos y sólidos hiperelásticos no lineales. No obstante, otros modelos de material para el sólido podrían ser fácilmente considerados puesto que los algoritmos desarrollados en esta tesis son ampliamente independientes del material particular utilizado para el sólido.

- Las tres incógnitas de nuestra formulación son la velocidad Euleriana, la presión Euleriana y el desplazamiento Lagrangiano. Estas tres incógnitas están asociadas con la ecuación de balance de momento lineal, la ecuación de conservación de masa y la ecuación cinemática que relaciona el desplazamiento Lagrangiano con la velocidad Euleriana. Las dos primeras ecuaciones se resuelven usando una técnica multiescala varacional y la última se resuelve usando colocación isogeométrica, lo cual lleva a un método híbrido que combina un enfoque variacional con un enfoque de colocación.

- Con respecto a la discretización espacial, una malla Euleriana se define en todo el dominio computacional y una malla Lagrangiana se ajusta a la región ocupada por cada sólido. Estas mallas son inicialmente construidas utilizando NURBS, que pueden ser utilizadas en dos y tres dimensiones, pero no permiten realizar refinamiento local tipo h. A continuación, los algoritmos son generalizados para utilizar ASTS como funciones base, las cuales soportan refinamiento local tipo h, pero sólo están definidas en dos dimensiones por el momento. Con respecto a la discretización temporal y estrategia de acoplamiento, utilizamos una técnica en bloque iterativa basada en el método α-generalizado, lo cual nos permite obtener un método completamente implícito y con precisión de segundo orden. Validamos la metodología comparando la velocidad terminal de un sólido en caída libre calculada en nuestras simulaciones con su valor teórico, el valor teórico se obtiene con total exactitud si las mallas utilizadas son suficientemente finas.

- La transferencia de datos entre la malla Euleriana y las mallas Lagrangianas se realiza sin introducir funciones delta de Dirac discretizadas. Las funciones base de las mallas se utilizan para transferir dichos datos. Además, procedemos de tal manera que no se necesita ningún tipo de proyección, evitando así los errores asociados con dicha proyección.

- Los polinomios de Lagrange, que son las funciones base estándar en el método de elementos finitos, tienen continuidad C^0 entre elementos para todos los órdenes. Por el contrario, NURBS y ASTS son capaces de obtener continuidad C^{p-1} entre elementos, lo cual tiene al menos dos importantes ventajas en nuestros métodos inmersos. En primer lugar, la alta continuidad entre elementos de las mallas Lagrangianas permite soportar deformaciones más grandes en estas mallas, aumentando así la robustez de nuestros algoritmos. En segundo lugar, la alta continuidad entre elementos de la malla Euleriana alivia el problema de reglas de cuadratura subóptima que aparecen en métodos inmersos para interacción fluido-estructura.

- Enfatizamos en la importancia de satisfacer adecuadamente la condición de incompresibilidad cerca de la interfaz fluido-sólido. Creemos que este problema requiere más atención en la comunidad de métodos inmersos para interacción fluido-estructura ya que puede dar lugar a errores no despreciables en simulaciones.

- Nuestro código basado en NURBS fue desarrollado sobre PetIGA [36, 22, 123], PetIGA añade funciones para trabajar con NURBS a la librería cientfica PETSc. Como paso inicial, un código serial fue desarrollado para validar la formulación [8, 7]. Luego, el código fue extendido para poder realizar cálculos en paralelo utilizando MPI. El último paso es no trivial puesto que estos métodos tienen varias mallas Lagrangianas moviéndose libremente por encima de una malla Euleriana y estas mallas tienen que comunicarse entre sí. Se ha probado que PetIGA es muy eficiente [36] y es capaz de mantener la alta continuidad de las funciones base entre procesadores. Sin embargo, todas las estructuras

de datos utilizadas para ensamblar el sistema están desarrolladas para mallas estructuradas. Por lo tanto, si la malla es no estructurada como en el caso de ASTS, PetIGA no ofrece ningún tipo de apoyo al usuario. En consecuencia, para todas las simulaciones utilizando ASTS presentadas en esta tesis, un código serial fue desarrollado directamente sobre PETSc sin utilizar PetIGA.

Los métodos inmersos de interacción fluido-estructura basados en splines presentados en esta tesis pueden ser aplicados a una amplia variedad de problemas en biomecánica y otros campos. En esta tesis, nos hemos centrado en flujo sanguíneo a escala celular. Los principales resultados de simulaciones son los siguientes:

- Cápsulas hiperelásticas, modeladas utilizando elementos NURBS, son utilizadas como imitadores celulares de glóbulos rojos. Siempre que la condición de incompresibilidad sea adecuadamente impuesta cerca de la interfaz fluido-sólido, nuestras cápsulas experimentan los mismos movimientos y formas que se han encontrado para glóbulos rojos en experimentos. Además, varias propiedades reológicas de la sangre son reproducidas en nuestras simulaciones.

- Modelos para células con núcleo tales como, por ejemplo, glóbulos blancos y células cancerosas en circulación, están actualmente mucho menos desarrollados que los modelos para glóbulos rojos. En esta tesis, presentamos cápsulas compuestas hiperelásticas para modelar este tipo de células. El comportamiento de las cápsulas compuestas en flujo de corte se estudia en dos dimensiones. Con el objetivo de analizar la influencia del núcleo, consideramos cápsulas compuestas con diferentes tamaños de inclusión y la misma membrana. Consistentemente con los resultados para vesículas compuestas [120], a medida que el tamaño de la inclusión aumenta, el ángulo de inclinación y la frecuencia de "tank-treading" de la cápsula compuesta disminuyen. Adicionalmente, aprovechándonos de la flexibilidad geométrica de nuestros métodos inmersos, estudiamos cómo una célula cancerosa en circulación es capaz de pasar por estrechamientos de canal. La simulación revela que la deformación más alta de la célula cancerosa en circulación y la velocidad más alta de fluido surgen cuando la célula sale del estrechamiento.

E.5 Futuras líneas de trabajo

El trabajo realizado en esta tesis deja varias líneas de investigación para el futuro:

- Hay varios fenómenos interesantes tales como, por ejemplo, marginación de células con núcleo en flujos parabólicos y separación de células con núcleo a partir de muestras de sangre, que implican la interacción de glóbulos rojos con glóbulos blancos o células cancerosas en circulación. En algunos de estos fenómenos, los efectos inerciales juegan un papel importante [59, 129]. Nuestros métodos

inmersos tienen en cuenta efectos inerciales los cuales tienen que ser despreciados en otros tipos de métodos inmersos como aquellos que se basan en el método de integral de contorno [49]. Por lo tanto, nuestra formulación parece muy adecuada para afrontar estas aplicaciones y esperamos hacerlo en el futuro próximo.

- Diferentes modelos pueden ser considerados para representar la membrana de las células. Por ejemplo, vesículas, las cuales están basadas en una representación de superficie, necesitan calcular la curvatura y por tanto necesitan evaluar derivadas segundas. El uso de NURBS o ASTS para representar estas superficies sería muy apropiado ya que son capaces de construir espacios de funciones con continuidad global C^1 o superior. Además, parece que una comparación precisa entre vesículas y cápsulas para establecer las principales diferencias en su comportamiento no existe en la literatura.

- B-splines que satisfacen la condición de incompresibilidad punto a punto [44] pueden ser consideradas para resolver la ecuación de balance de momento lineal y la ecuación de conservación de masa en lugar de usar la técnica VMS. En general, esta alternativa resulta atractiva para aplicaciones con números de Reynolds bajos, lo cual es el caso de flujo sanguíneo a escala celular. Adicionalmente, esta formulación podría ayudar a satisfacer la condición de incompresibilidad cerca de la interfaz fluido-sólido. Sin embargo, la convergencia del residuo, la cual es siempre complicada en problema del tipo punto de silla como el de [44], será todavía más difícil de obtener a medida que sólidos son inmersos en el dominio.

- Desarrollar mallas Eulerianas basadas en splines con capacidad para realizar refinamiento local adaptativo de tipo h con el objetivo de aumentar la resolución cerca de la interfaz fluido-sólido es una dirección de trabajo prometedora.

F Extended summary in Galician

F.1 Introdución

Nas últimas décadas, o desenvolvemento de metodoloxías de modelación e simulación para problemas de interacción fluído-estrutura atraeu enorme atención na comunidade da mecánica computacional. Un método computacional preciso, xeométricamente flexible, robusto, e eficiente é crucial á hora de resolver complicados problemas de interacción fluído-estrutura, os cales están presentes en enxeñaría biomédica [15, 16], enxeñaría aeroespacial [116, 117], enxeñaría civil [86, 118, 98, 45], enxeñaría mariña [130, 127] e enxeñaría eólica [12, 13] por nomear uns exemplos. Até a data, non hai un método de interacción fluído-estrutura que poida ser utilizado en todos os tipos de problemas de interacción fluído-estrutura. Por tanto, diferentes tipos de métodos de interacción fluído-estrutura foron desenvoltos para aplicacións de interacción fluído-estrutura con distintas características.

Os métodos de interacción fluído-estrutura baseados en mallas poden ser clasificados en dous grupos principais, en concreto, métodos axustados ao contorno e métodos inmersos. Nos métodos axustados ao contorno, unha malla axustada ao dominio do fluído utilizando a descripción arbitraria de Euler-Lagrange e outra malla é axustada ao dominio do sólido utilizando a descripción Lagrangiana, estas dúas mallas son habitualmente conformes na interface fluído-estrutura de maneira que a imposición de condicións cinemáticas poida ser realizada de forma exacta. A malla do fluído ten que ser actualizada en cada paso de tempo resolvendo un problema de valores de contorno onde as condicións de Dirichlet aplicadas son os desprazamentos do sólido na interface. A medida que o se move o sólido, a calidade da malla do fluído diminuirá e acabará chegando un momento onde o fluído necesitará ser remallado. Para unha explicación minuciosa dos métodos axustados ao contorno e exemplos de aplicación, o lector interesado pode consultar [17]. En métodos inmersos, unha malla Euleriana abarca todo o dominio computacional e unha malla Lagrangiana é axustada ao dominio do sólido. Por conseguinte, a necesidade de remallar a malla axustada ao dominio do fluído desapareceu. Non obstante, obtense esta vantaxe ao prezo de asumir que o campo da presión é continuo na interface sólido-fluído e isto é especialmente problemático cando se consideran sólidos moi delgados [67].

O movemento do sólido é o factor clave para decidir se utilizar un método axustado ao contorno ou un método inmerso. Se utilizamos un método axustado ao contorno nunha aplicación onde o sólido experimenta grandes translacións ou rotacións, a malla axustada ao dominio do fluído distorsionarase moito cada poucos pasos de tempo, afectando a precisión da simulación. O remallado pode ser utilizado para combater este problema, pero, ademais do seu alto custo computacional, necesitaranse proxeccións da velocidade e da presión entre a antiga e a nova malla, o cal introduce imprecisións na simulación. Estas dificultades amplifícanse cando hai varios sólidos e estes sólidos móvense por todo o dominio, por exemplo, en fluxos de partículas [52]. Ademais, nos casos onde a topolox´a da rexión ocupada polo fluído vai cambiar durante a simulación debido ao contacto entre diferentes sólidos, a aplicación dun método axustado ao contorno é aínda un problema aberto. En todas as situacións mencionadas anteriormente, os métodos inmersos emerxen

como unha alternativa adecuada aos métodos axustados ao contorno posto que as mallas Lagrangianas asociadas a cada sólido poden moverse libremente por riba da malla Euleriana.

Os dous principais obxectivos desta tese son o desenvolvemento de métodos inmersos para interacción fluído-estrutura usando splines e aplicar devanditos métodos ao fluxo sanguíneo a escala celular, o cal é un dos fluxos de partículas que recentemente recibiu máis atención debido á súa conexión con diferentes enfermidades como o cancro [77]. Funcións spline desenvoltas no campo do deseño asistido por computador como, por exemplo, NURBS [106] e T-splines [113], foron empregadas por primeira vez en mecánica computacional en [63] e [10], respectivamente. A alta continuidade entre elementos, a non negatividade e flexibilidade xeométrica das funcións spline orixinaron importantes avances tanto na mecánica de fluídos [2, 11] como na mecánica de sólidos [38, 81]. Por tanto, a aplicación de splines a métodos inmersos de interacción fluído-estrutura baseados en mallas parece unha forma interesante de mellorar a precisión, a robustez e a flexibilidade xeométrica das versións actuais dos devanditos métodos.

Esta tese estende a estado da arte en métodos inmersos de interacción fluído-estrutura baseados en mallas desenvolvendo os avances que se resumen a continuación:

1. Un método inmerso baseado en NURBS para interacción fluído-estrutura. Presentamos unha formulación híbrida que combina un enfoque variacional cun enfoque de colocación, inmersa e completamente implícita para interacción fluído-estrutura. Esta técnica traballa con fluídos que son viscosos e incompresibles e con sólidos que son non lineais e incompresibles. O análise isoxeométrico baseado en NURBS empregase para a discretización espacial e o método α-xeneralizado empregase para a discretización temporal. A transferencia de datos entre a malla Euleriana e as mallas Lagrangianas é levada a cabo usando tamén as funcións NURBS, o cal evita o uso das denominadas deltas de Dirac discretizadas que son a miúdo empregadas en métodos inmersos. Validamos o método comparando as nosas solucións numéricas con datos teóricos para velocidades de corpos ríxidos en caída libre en dúas e tres dimensións. Ver a Sección 4 e o Apéndice A para detalles adicionais.

2. Colocación isoxeométrica baseada en T-splines adecuadas para análise de orde arbitraria. Co obxectivo de estender o noso método inmerso de interacción fluído-estrutura a T-splines, tivemos que desenvolver primeiro métodos de colocación isoxeométrica baseados en T-splines adecuadas para análise de orde arbitraria, o cal representaba un problema aberto en mecánica computacional. T-splines adecuadas para análise permiten realizar refinamento de tipo h ao mesmo tempo que manteñen tanto o custo de só unha avaliación puntual por grao de liberdade como os índices de converxencia dos métodos de colocación baseados en NURBS. Ademais, T-splines adecuados para análise (a partir de agora denominadas ASTS polas súas iniciais en inglés) son capaces de crear mallas altamente non uniformes sen introducir elementos con cocientes de forma altos, o cal evita inestabilidades numéricas que aparecen en problemas con regularidade reducida cando condicións de contorno de Neumann son

impostas en forma forte en mallas con elementos con cocientes de forma altos. Ver a Sección 5 e o Apéndice B para detalles adicionais.

3. Un método inmerso baseado en ASTS para interacción fluído-estrutura. Os nosos métodos inmersos para interacción fluído-estrutura son xeneralizados para traballar con ASTS. Validamos os nosos resultados numéricos comparándoos con solucións teóricas. ASTS posúen dúas vantaxes principais cando son comparadas con NURBS, en concreto, ASTS permiten realizar refinamento local de tipo *h* e son desestructuradas. Estas propiedades teñen o potencial de aumentar a eficiencia e flexibilidade de discretizacións espaciais baseadas en NURBS. Ver a Sección 6 e o Apéndice C para detalles adicionais.

4. Cápsulas e cápsulas compostas como imitadores celulares en fluxo. Realizamos simulacións con cápsulas en fluxos de corte e fluxos parabólicos co objectivo de reproducir os principais movementos e formas que adquiren os glóbulos vermellos nestes fluxos. Tamén medidos e comparamos con experimentos varias propiedades reolóxicas do sangue en pequenas escalas. Ademais, presentamos o concepto de cápsula composta como un imitador celular para células con núcleo como glóbulos brancos e células cancerosas en circulación. Realizamos simulacións con cápsulas compostas hiperelásticas en fluxo de corte co obxectivo de estudar o seu comportamento en dúas e tres dimensións. Finalmente, mostramos a habilidade das células cancerosas en circulación para atravesar un estreitamento dunha canle, esta propiedade está actualmente sendo empregada para separar células cancerosas en circulación en mostras de sangue [134]. Ver a Sección 7 e o Apéndice C para detalles adicionais.

F.2 Obxectivos

Os obxectivos desta tese poden ser clasificados en tres categorías:

- Desenvolvemento e validación de métodos numéricos. O obxectivo é idear métodos inmersos para interacción fluído-estrutura baseados en NURBS e ASTS. Devanditos métodos deberán aproveitarse da alta continuidade entre elementos das funcións spline co fin de aumentar a precisión e robustez con respecto a anteriores métodos inmersos como o método de elementos finitos inmersos [131]. Ademais, o uso de NURBS e ASTS permitiría a representación exacta de curvas cónicas e superficies cuadráticas que aparecen a miúdo en problemas de interacción fluído-estrutura. Os métodos propostos deben ser validados resolvendo problemas con solución teórica coñecida.

- Aplicación a biomecánica. O obxectivo é estudar o comportamento de diferentes tipos de células sanguíneas en fluxo sanguíneo en pequenas escalas. Varios autores mencionaron que os polinomios de Lagrange non son capaces de manexar as grandes deformacións dos glóbulos vermellos en fluxo de corte e fluxo parabólico [90, 91], propoñendo métodos alternativos baseados en partículas [88, 87].

A maior robustez das funcións NURBS é utilizada para superar este problema. Co fin de avaliar os resultados das nosas simulacións, estes deben ser comparados con experimentos de glóbulos vermellos. Ademais, pretendemos desenvolver modelos para células con núcleo como os glóbulos brancos e as células cancerosas en circulación. Finalmente, exemplos en xeometrías non triviais deberían ser considerados para ilustrar a flexibilidade xeométrica dos nosos métodos inmersos.

- Implementación dos algoritmos propostos. A maioría dos métodos inmersos para interacción fluído-estrutura baseados en mallas foron só implementados en códigos seriais. Nesta tese, o obxectivo é ir un paso máis aló e realizar implementacións en paralelo dos nosos algoritmos.

F.3 Definición do problema a resolver

A continuación procédese a enunciar o problema matemático a resolver nesta tese. Sexa $d = \{2, 3\}$ o número de dimensións espaciais. Sexa Ω_t^f e $\Omega_t^s \subset \mathbb{R}^d$ dous conxuntos abertos que representan os dominios ocupados polo fluído viscoso incompresible e o sólido hiperelástico incompresible, respectivamente, con Γ_t^f e Γ_t^s os seus correspondentes contornos. Estas dúas rexións teñen parte do seu contorno en común, a este contorno denominarémolo interface fluído-sólido Γ_t^I ($\Gamma_t^I = \overline{\Omega_t^f} \cap \overline{\Omega_t^s}$). Sexa Sea $\Omega \subset \mathbb{R}^d$ un conxunto aberto que representa o dominio ocupado por tanto o fluído como o sólido, é dicir, $\overline{\Omega} = \overline{\Omega_t^f \cup \Omega_t^s}$. Sexa Γ o contorno de Ω. Sexan $\boldsymbol{v} : \Omega \times (0, T) \mapsto \mathbb{R}^d$, $p : \Omega \times (0, T) \mapsto \mathbb{R}$ e $\boldsymbol{u} : \Omega_0^s \times (0, T) \mapsto \mathbb{R}^d$ a velocidade Euleriana, a presión Euleriana e o desprazamento Lagrangiano, respectivamente, onde $(0, T)$ é o intervalo temporal de interese e Ω_0^s é a configuración de referencia de Ω_t^s. Sexan $\boldsymbol{X} \in \Omega_0^s$, $\boldsymbol{x} \in \Omega$, $\boldsymbol{x} \in \Omega$ e $\boldsymbol{\varphi} : \Omega_0^s \times (0, T) \mapsto \Omega_t^s$ un punto material, un punto espacial, e o mapeo $\boldsymbol{\varphi}(\boldsymbol{X}, t) = \boldsymbol{X} + \boldsymbol{u}(\boldsymbol{X}, t)$, respectivamente.

As tres ecuacións que definen o noso problema son a ecuación de balance de momento lineal, a ecuación de conservación de masa e a relación entre o desprazamento Lagrangiano e a velocidade Euleriana:

$$\rho^f \left(\frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v} \cdot \nabla_{\boldsymbol{x}} \boldsymbol{v} \right) = \nabla_{\boldsymbol{x}} \cdot \boldsymbol{\sigma}^f + \rho^f \boldsymbol{g} + \boldsymbol{\mathcal{F}} \qquad \text{in} \quad \Omega \times (0, T) \,, \tag{44}$$

$$\nabla_{\boldsymbol{x}} \cdot \boldsymbol{v} = 0 \qquad \text{in} \quad \Omega \times (0, T) \,, \tag{45}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} = \boldsymbol{v} \qquad \text{in} \quad \Omega_t^s \times (0, T) \,, \tag{46}$$

con:

$$\boldsymbol{\mathcal{F}} = \begin{cases} 0, & \boldsymbol{x} \in \Omega_t^f \\ \left(\rho^f - \rho^s\right)\left(\frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v} \cdot \nabla_{\boldsymbol{x}} \boldsymbol{v} - \boldsymbol{g}\right) + \nabla_{\boldsymbol{x}} \cdot \left(\boldsymbol{\sigma}^s - \boldsymbol{\sigma}^f\right), & \boldsymbol{x} \in \Omega_t^s \end{cases} , \tag{47}$$

$$\boldsymbol{\sigma}^f = -p\boldsymbol{I} + 2\mu \nabla_{\boldsymbol{x}}^{\text{sym}} \boldsymbol{v} \,, \tag{48}$$

$$\boldsymbol{\sigma}^s = -p\boldsymbol{I} + \boldsymbol{F}\boldsymbol{S}\boldsymbol{F}^T / J \,, \tag{49}$$

onde ρ^f e ρ^s son a densidade do fluído e do sólido, respectivamente, $\boldsymbol{\sigma}^f$ e $\boldsymbol{\sigma}^s$ son o tensor de tensións de Cauchy do fluído e do sólido, respectivamente, $\boldsymbol{g}$ é unha forza de volume externa actuando no sistema,

A denota o tensor identidade $\mathbb{R}^{d\times d}$, μ é a viscosidade dinámica do fluído, $\nabla_{\boldsymbol{x}}^{\text{sym}}(\cdot)$ é o operador gradiente simétrico definido como $\nabla_{\boldsymbol{x}}^{\text{sym}}\boldsymbol{v} = (\nabla_{\boldsymbol{x}}\boldsymbol{v} + \nabla_{\boldsymbol{x}}\boldsymbol{v}^T)/2$, $\boldsymbol{F} : \Omega_0^s \times (0,T) \mapsto \mathbb{R}^{d\times d}$ é o gradiente de deformación definido como $\boldsymbol{F} = \nabla_{\boldsymbol{X}}\boldsymbol{\varphi} = \boldsymbol{I} + \nabla_{\boldsymbol{X}}\boldsymbol{u}$, $\boldsymbol{S}$ é o segundo tensor de tensións de Piola-Kirchhoff do material considerado e $J = \det(\boldsymbol{F})$ é o determinante do Jacobiano o cal é exactamente igual a 1 a nivel continuo xa que o sólido é incompresible.

Adecuadas condicións de contorno e iniciais deben ser impostas ás ecuacións anteriores para obter un problema de interacción fluído-estrutura ben definido. Co obxectivo de impor condicións de contorno, dividimos o contorno Γ en dous conxunto $\mathcal{B}^s = \Gamma \cap \Gamma_t^s$ e $\mathcal{B}^f = \Gamma \cap \Gamma_t^f$ de maneira que $\Gamma = \overline{\mathcal{B}^s \cup \mathcal{B}^f}$ e $\emptyset = \mathcal{B}^s \cap \mathcal{B}^f$. é importante darse conta que $\mathcal{B}^s \neq \Gamma_t^s$ posto que polo menos parte do sólido atópase inmerso no fluído. Na interface Γ_t^I, dúas condicións de contorno son impostas, en concreto, igualdade de velocidades e compatibilidade de tensións entre o fluído e o sólido. A igualdade de velocidades imponse por medio da ecuación (40) e a compatibilidade de tensións imponse na forma débil da ecuación (38). En $\mathcal{B}^f$, distinguimos entre condicións de contorno de Dirichlet en $\mathcal{B}_D^f$ e condicións de contorno de Neumann en $\mathcal{B}_N^f$ de maneira que $\mathcal{B}^f = \overline{\mathcal{B}_D^f \cup \mathcal{B}_N^f}$ e $\mathcal{B}_D^f \cap \mathcal{B}_N^f = \emptyset$. En $\mathcal{B}^s$, consideramos condicións de contorno Dirichlet homoxéneas. Con respecto ás condicións iniciais, a posición inicial do sólido, o desprazamento Lagrangiano e a velocidade Euleriana deben ser especificadas.

F.4 Contribucións da tese

Esta sección resume as principais contribucińs e ideas desta tese. Esta tese propón métodos inmersos baseados en mallas para interacción fluído-estrutura coas seguintes características:

- Asúmese que tanto o fluído como o sólido son incompresibles, o cal é un caso de especial interese en varios problemas de biomecánica [50]. Ademais diso, traballamos con fluídos viscosos Newtonianos e sólidos hiperelásticos non lineales. Non obstante, outros modelos de material para o sólido poderían ser fácilmente considerados posto que os algoritmos desenvoltos nesta tese son amplamente independentes do material particular empregado para o sólido.

- As tres incógnitas da nosa formulación son a velocidade Euleriana, a presión Euleriana e o desprazamento Lagrangiano. Estas tres incógnitas están asociadas coa ecuación de balance de momento lineal, a ecuación de conservación de masa e a ecuación cinemática que relaciona o desprazamento Lagrangiano coa velocidade Euleriana. As dúas primeiras ecuacións resólvense usando unha técnica multiescala varacional e a última resólvese usando colocación isoxeométrica, o cal leva a un método híbrido que combina un enfoque variacional cun enfoque de colocación.

- Con respecto á discretización espacial, unha malla Euleriana defínese en todo o dominio computacional e unha malla Lagrangiana axústase á rexión ocupada por cada sólido. Estas mallas son inicialmente construídas empregando NURBS, que poden ser empregadas en dúas e tres dimensións, pero

non permiten realizar refinamento local de tipo h. A continuación, os algoritmos son xeneralizados para empregar ASTS como funciones base, as cales soportan refinamento local de tipo h, pero só están definidas en dúas dimensións polo momento. Con respecto á discretización temporal e estratexia de axuste, utilizamos unha técnica en bloque iterativa baseada no método α-xeneralizado, o cal nos permite obter un método completamente implícito e con precisión de segunda orde. Validamos a metodoloxía comparando a velocidade terminal dun sólido en caída libre calculada nas nosas simulacións co seu valor teórico, o valor teórico obtense con total exactitude se as mallas empregadas son suficientemente finas.

- A transferencia de datos entre a malla Euleriana e as mallas Lagrangianas realízase sen introducir funciones delta de Dirac discretizadas. As funcións base das mallas empréganse para transferir devanditos datos. Ademais, procedemos de tal maneira que non se necesita ningún tipo de proxección, evitando así os erros asociados coa devandita proxección.

- Os polinomios de Lagrange, que son as funcións base estándar no método de elementos finitos, teñen continuidade C^0 entre elementos para todas as ordes. Pola contra, NURBS e ASTS son capaces de obter continuidade C^{p-1} entre elementos, o cal ten polo menos dúas importantes vantaxes nos nosos métodos inmersos. En primeiro lugar, a alta continuidade entre elementos das mallas Lagrangianas permite soportar deformacións máis grandes nestas mallas, aumentando así a robustez dos nosos algoritmos. En segundo lugar, a alta continuidade entre elementos da malla Euleriana alivia o problema das regras de cadratura subóptima que aparecen en métodos inmersos para interacción fluído-estrutura.

- Destacamos a importancia de satisfacer adecuadamente a condición de incompresibilidade preto da interface fluído-sólido. Cremos que este problema require máis atención na comunicade de métodos inmersos para interacción fluído-estrutura xa que pode dar lugar a erros importantes en simulacións.

- O noso código baseado en NURBS foi desenvolvido sobre PetIGA [36, 22, 123], PetIGA engade funcións para traballar con NURBS á libraría científica PETSc [8, 7]. Como paso inicial, un código serial foi desenvolto para validar a formulación. Logo, o código foi estendido para poder realizar cálculos en paralelo empregando MPI. O derradeiro paso non é trivial posto que estes métodos teñen varias mallas Lagrangianas movéndose libremente por riba dunha malla Euleriana e estas mallas teñen que comunicarse entre si. Probouse que PetIGA é moi eficiente [36] e é capaz de manter a alta continuidade das funcións base entre procesadores. Con todo, todas as estruturas de datos utilizadas para ensamblar o sistema están desenvoltas para mallas estruturadas. Por tanto, si a malla é non estruturada como no caso de ASTS, PetIGA non ofrece ningún tipo de apoio ao usuario. En consecuencia, para todas as simulacións empregando ASTS presentadas nesta tese, un código serial foi desenvolto directamente sobre PETSc sen empregar PetIGA.

Os métodos inmersos de interacción fluído-estrutura baseados en splines presentados nesta tese poden ser aplicados a unha ampla variedade de problemas en biomecánica e outros campos. Nesta tese, centrámonos en fluxo sanguíneo a escala celular. Os principais resultados de simulacións son os seguintes:

- Cápsulas hiperelásticas, modeladas utilizando elementos NURBS, son empregadas como imitadores celulares de glóbulos vermellos. Sempre que a condición de incompresibilidade sexa adecuadamente imposta preto da interface fluído-sólido, as nosas cápsulas experimentan os mesmos movementos e formas que se atoparon para glóbulos vermellos en experimentos. Ademais, varias propiedades reolóxicas do sangue son reproducidas nas nosas simulacións.

- Modelos para células con núcleo como, por exemplo, glóbulos brancos e células cancerosas en circulación, están actualmente moito menos desenvoltos cos modelos para glóbulos vermellos. Nesta tese, presentamos cápsulas compostas hiperelásticas para modelar este tipo de células. O comportamento das cápsulas compostas en fluxo de corte estúdase en dúas dimensiúns. Co obxectivo de analizar a influencia do núcleo, consideramos cápsulas compostas con diferentes tamaños de inclusión e a mesma membrana. Consistentemente cos resultados para vesículas compostas [120], a medida que o tamaño da inclusión aumenta, o ángulo de inclinación e a frecuencia de "tank-treading" da cápsula composta diminúen. Adicionalmente, aproveitándonos da flexibilidade xeométrica dos nosos métodos inmersos, estudamos como unha célula cancerosa en circulación é capaz de pasar polo estreitamento dunha canle. A simulación revela que a deformación máis alta da élula cancerosa en circulación e a velocidade máis alta de fluído xorden cando a célula sale do estreitamento.

F.5 Futuras liñas de traballo

O traballo realizado nesta tese deixa varias liñas de investigación para o futuro:

- Hai varios fenómenos interesantes como, por exemplo, marxinación de células con núcleo en fluxos parabólicos e separación de células con núcleo a partir de mostras de sangue, que implican a interacción de glóbulos vermellos con glóbulos brancos ou células cancerosas en circulación. Nalgúns destes fenómenos, os efectos inerciales xogan un papel importante [59, 129]. Os nosos métodos inmersos teñen en conta efectos inerciales os cales teñen que ser desprezados noutros tipos de métodos inmersos como aqueles que se basean no método de integral de contorno [49]. Por tanto, a nosa formulación parece moi adecuada para afrontar estas aplicacións e esperamos facelo no futuro próximo.

- Diferentes modelos poden ser considerados para representar a membrana das células. Por exemplo, vesículas, as cales están baseadas nunha representación de superficie, necesitan calcular a curvatura e por tanto necesitan avaliar derivadas segundas. O uso de NURBS ou ASTS para representar estas superficies sería moi apropiado xa que son capaces de construír espazos de funcións con continuidade

global C^1 ou superior. Ademais, parece que unha comparación precisa entre vesícuas e cápsulas para establecer as principais diferenzas no seu comportamento non existe na literatura.

- B-splines que satisfán a condición de incompresibilidade punto a punto [44] poden ser consideradas para resolver a ecuación de balance de momento lineal e a ecuación de conservación de masa en lugar de usar a técnica VMS. En xeral, esta alternativa resulta atractiva para aplicacións con números de Reynolds baixos, o cal é o caso do fluxo sanguíneo a escala celular. Adicionalmente, esta formulación podería axudar a satisfacer a condición de incompresibilidade preto da interface fluído-sólido. Non obstante, a converxencia do residuo, a cal é sempre complicada en problemas do tipo punto de cadeira como en [44], será aínda máis difícil de obter a medida que sólidos son inmersos no dominio.

- Desenvolver mallas Eulerianas baseadas en splines con capacidade para realizar refinamento local adaptativo de tipo h co obxectivo de aumentar a resolución preto da interface fluído-sólido é unha dirección de traballo prometedora.